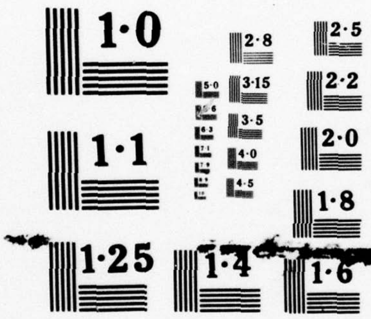
AD-A069 665 ARCON CORP WALTHAM MA F/G 9/2
MATHEMATICAL MODELLING.(U)
APR 79 S WOOLF, E COHEN F19628-76-C-0247
UNCLASSIFIED RADC-TR-79-65 NL

1 OF 2
AD
A069665



NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

RADC-TR-79-66
Final Technical Report
April 1979

MATHEMATICAL MODELLING

ARCON Corporation

Stanley Woolf
Edward Cohen

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-79-65 has been reviewed and is approved for publication.

APPROVED: John C. Garth

JOHN C. GARTH
Project Engineer

APPROVED: Robert M Barrett

ROBERT M. BARRETT
Director, Solid State Sciences Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

(19) REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
(18) RADC-TR-79-65

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
(6) MATHEMATICAL MODELLING,

5. TYPE OF REPORT & PERIOD COVERED
(9) Final Technical Report,
1 Sep 76 — 30 Apr 78,

6. PERFORMING ORG. REPORT NUMBER
N/A

7. AUTHOR(s)
(10) Stanley Woolf
Edward Cohen

8. CONTRACT OR GRANT NUMBER(s)
(15) F19628-76-C-0247 new

9. PERFORMING ORGANIZATION NAME AND ADDRESS
ARCON Corporation
260 Bear Hill Road
Waltham MA 02154

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
61102F, 62702F
(16) 2306J230 (17) J2

11. CONTROLLING OFFICE NAME AND ADDRESS
Deputy for Electronic Technology (RADC/ESR)
Hanscom AFB MA 01730

12. REPORT DATE
(11) April 1979

13. NUMBER OF PAGES
~~187~~ (12) 185p.

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
Same

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
N/A

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
Same

18. SUPPLEMENTARY NOTES
RADC Project Engineer: John C. Garth (ESR)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Electron Transport
Surface Acoustic Wave Filters
Phase-Constrained Guided Modes in Dielectric Slabs

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In this report we will describe the techniques and computer programs developed to analyze various problems. The scope of our work extended into three technical areas:

1) the problem of electron scattering and transport in solids;
2) the design of surface acoustic wave devices, and
3) the study of phase-constrained guided modes in dielectric slabs.

403370

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

PREFACE

The analysis, computer programs and results discussed herein are the result of analytical research performed for

Rome Air Development Center
Deputy for Electronic Technology
L. G. Hanscom Air Force Base, Massachusetts 01731

EVALUATION

1. This report is the Final Report on Contract F19628-76-C-0247 with ARCON Corporation. It covers research done on mathematical modelling during the twenty month period 1 September 1976 to 30 April 1978. The objective of the research is the development of new mathematical solution techniques for application in the areas of kilovolt electron transport in solids, surface acoustic wave device design, and mode propagation in optical waveguides. Significant progress was made in solving the electron transport equation as an alternative to Monte Carlo calculations, and new methods for (a) design of 3-phase SAW transducers and (b) tailoring the refractive index profile in an optical waveguide to obtain specified propagation characteristics were developed.

2. The above work is of value since it provides new computational tools for improving the radiation hardness of electronic devices and for developing advanced devices for USAF radar, communication and ECM systems.

John C. Garth

JOHN C. GARTH
Project Engineer

1. INTRODUCTION

During the past twenty months our work consisted of technical efforts in the areas of electron transport in scattering media, surface acoustic wave device design and the determination of phase-constrained guided optical modes in dielectric slabs. These efforts resulted in the construction of mathematical models and accompanying computer programs which will be described in detail in the following sections.

2. Electron Transport in Solids

2.1 Introduction

A detailed description of the behavior of electrons as they are transported through a scattering medium can be obtained by solving the Spencer-Lewis equation. The Spencer-Lewis[1] equation has the same mathematical form as the time-dependent Boltzmann transport equation. In it the electron energy and total electron path length are related through the continuous slowing-down approximation. The quantity which emerges as the solution of the Spencer-Lewis equation is the electron or particle flux which is a function of the particle's position, direction of motion and total trajectory length. Assuming a slab geometry (Fig. 1-1), an assumption that will be adopted throughout this document, let the electron flux be denoted by $f(x,\mu,s)$, where x is the position along the normal to the slab surface, μ is the cosine of the polar angle along the direction of electron travel, and s is the electron path length ($= s(E)$, E = electron energy). Then the Spencer-Lewis equation for $f(x,\mu,s)$ has the form

$$\frac{\partial f}{\partial s} + \mu \frac{\partial f}{\partial x} = \frac{1}{\lambda(s)} \int_{-1}^{1} d\mu' \, P\,(\mu' \rightarrow \mu, s)\,[f(x,\mu',s) - f(x,\mu,s)] \tag{1}$$

where $\lambda(s)$ is the total energy-dependent electron mean-free path and is expressed here as a function of path length, s. The relation between the path length, s, and the electron energy, E, is given by the stopping power formula[1]

$$s(E) = \int_{E}^{E_{max}} dE \,/\, \left|\frac{dE}{ds}\,(E)\right| \quad , \tag{2}$$

where E_{max} is the initial electron energy, and dE/ds is the electron stopping power, a material dependent function of energy. An expression for the stopping power is given by the Berger-Seltzer[(2)] formula.

Of the remaining terms in the Spencer-Lewis equation, $P(\mu' \rightarrow \mu, s)$ is the path length or energy dependent probability density of scattering from angle μ' to angle μ

i.e. for conservative scattering

$$\int_{-1}^{1} P\ (\mu' \rightarrow \mu, s)\ d\mu' = 1 \quad , \tag{3}$$

and

$S(x, \mu, s)$ is the source term.

Full analytical solutions to the Spencer-Lewis equation are rare. They do not exist for situations of interest such as beam or monodirectional plane sources with either isotropic or anisotropic scattering but rather for such situations as a plane, uniform isotropic source with no scattering (particle streaming) and scattering of electrons emanating from a uniformly (spatially) distributed source in an infinite medium (gradient term absent).

There are a number of numerical solution methods for this equation. The method of moments, developed by Spencer[(3)], applies to the case of a plane monoenergetic source in an infinite medium. Garth[(4)] and O'Brien developed a finite-difference solution based on the method of discrete ordinates. Both of these methods incorporate anisotropic scattering kernels, a somewhat complicated relativistic coulomb scattering function in the former, and a less complicated screened Rutherford formula for the latter. The screened Rutherford scattering formula is used in the present work since it provides a simple means by which the anisotropy of the electron scattering may be simulated. The degree of anisotropy can be systematically varied with appropriate adjustment of the screening parameter, η. The screened-Rutherford expression for the angular scattering probability density is (assuming azimuthal symmetry),

$$P(\mu'\rightarrow\mu) = \frac{\eta(1+\frac{\eta}{2})\ (1+\eta-\mu\mu')}{[\eta^2+2\eta(1-\mu\mu')+(\mu-\mu')^2]^{3/2}} \qquad (4)$$

where μ' and μ are, respectively, the pre and post-collision electron velocity direction cosines w. r. t. the slab axis, x (see Fig. 1) and η is the Rutherford screening factor. The screening factor is energy and material dependent. That is, for a given material of atomic number Z

$$\eta(E) = \frac{3.2 \times 10^{-3}}{E(KeV)} Z^{2/3} \quad . \qquad (5)$$

Another method for the numerical solution of the Spencer-Lewis equation, the P_ℓ method, will be reported. A computer code was developed which obtains solutions of this equation in terms of Legendre expansions of the electron flux. A parallel effort also took place in which orders-of-scattering solutions of the Spencer-Lewis equation were obtained in analytic form for the first three orders in the case of isotropic scattering and the first two orders for screened Rutherford scattering. These analytic solutions were then used as input to a recursion formula, with which accurate determinations of higher order scattering solutions were obtained numerically. The orders-of-scattering solutions were then used to test the validity of the P_ℓ computer code.

2.2 The P_ℓ Method

The principle of operation of the P_ℓ method consists of expanding the electron flux, source and scattering functions in Legendre series in angle and then solving the resulting partial differential equations for the Legendre series coefficients of the electron flux. Implementation of this method has been previously accomplished by researchers in the nuclear reactor field. Their interest was primarily centered on steady-state neutron transport in weakly anisotropic scattering media. As pointed out by Clark and Hansen[(4)], the P_ℓ method can also be employed in the time-dependent transport case. The adaptation of the time-dependent P_ℓ method to the solution of the Spencer-Lewis equation is straightforward since there is a direct-correspondence between the path length variable, s, of the latter and the time variable, t, of the former.

2.2.1 Derivation of the Flux Coefficient Equations

Equation (1) may be re-written in the following form

$$\left(\frac{\partial}{\partial s} + \mu \frac{\partial}{\partial x} + \frac{1}{\lambda}\right) f(x,\mu,s) = \frac{1}{\lambda} \int_{-1}^{1} d\mu' \int_{0}^{2\pi} d\varphi \, P(\cos\alpha) \, f(x,\mu',s) + S(x,\mu,s) \qquad (6)$$

where cos α, the cosine of the deflection angle subtended by the initial and final directions of the scattered particle, is given by

$$\cos\alpha = \mu\mu' + \sqrt{(1-\mu^2)(1-\mu')^2} \, \cos\varphi \qquad (7)$$

(μ and μ' are as previously defined, and φ is the azimuthal deflection).

The particle flux, source function, and scattering probability are then expressed as Legendre series expansions in angle. That is,

$$f(x,\mu,s) \;=\; \sum_{\ell=0}^{\infty} (\ell + \tfrac{1}{2})\, f_\ell(x,s)\, P_\ell(\mu) \quad , \tag{8}$$

$$S(x,\mu,s) \;=\; \sum_{\ell=0}^{\infty} (\ell + \tfrac{1}{2})\, S_\ell(x,s)\, P_\ell(\mu) \quad , \tag{9}$$

$$P(\cos\alpha) \;=\; \sum_{\ell=0}^{\infty} (\ell + \tfrac{1}{2}\; d_\ell\, P_\ell\, (\cos\alpha) \quad , \tag{10}$$

where $P_\ell(z)$ is the Legendre polynomial in z of order ℓ, $f_\ell(x,s)$, $S_\ell(x,s)$ and d_ℓ are the coefficients of the ℓ-th order term of the respective Legendre series for the flux, source and scattering probability. Before these expansions are substituted into Eqn. (6), it is necessary to express the scattering probability in terms of the initial and final direction cosines, μ' and μ. The addition theorem for spherical harmonics states that

$$P_\ell(\cos\alpha) \;=\; P_\ell(\mu)\, P_\ell(\mu') \;+\; 2 \sum_{k=1}^{\ell} P_\ell^k(\mu)\, P_\ell^k(\mu')\, \frac{(\ell-k)!}{(\ell+k)!} \cos k\varphi \quad . \tag{11}$$

With the above identity the azimuthal portion of the scattering integral of Eqn. (6) may be evaluated as follows:

$$\int_0^{2\pi} d\varphi\, P(\cos\alpha) \;=$$

$$\sum_{\ell=0}^{\infty} (\ell+\tfrac{1}{2}) \int_0^{2\pi} d\varphi\, d_\ell \left[P_\ell(\mu)\, P_\ell(\mu') + 2 \sum_{k=1}^{\ell} P_\ell^{(k)}(\mu')\, \frac{(\ell-k)!}{(\ell+k)!} \cos k\varphi \right] \quad , \tag{12}$$

where the $P_\ell^{(k)}(\mu)$ are associated Legendre functions.

Simplification is possible since $\int_0^{2\pi} d\varphi \cos k\varphi = 0$; k an integer.

Therefore

$$\int_0^{2\pi} d\varphi\, P(\cos\alpha) = 2\pi \sum_{\ell=0}^{\infty} (\ell + \frac{1}{2})\, d_\ell P_\ell(\mu') \quad . \tag{13}$$

If $f(x, \mu', s)$ is expanded in like manner to that given by Eqn. (8), i.e.

$$f(x, \mu', s) = \sum_{j=0}^{\infty} (j + \frac{1}{2})\, f_j(x, s)\, P_j(\mu') \quad , \tag{14}$$

where $f_j(x, s)$ and $P_j(\mu)$ are defined in the usual manner (cf. Eqns. 8, 9, 10), then the right hand side of Eqn. (6) becomes, upon substitution of Eqns. (8, 9, 10, 14)

$$\begin{aligned}
&\frac{1}{\lambda} \int_{-1}^{1} d\mu' \int_0^{2\pi} d\varphi\, P(\cos\alpha)\, f(x, \mu', s) = \\
&\frac{2\pi}{\lambda} \sum_{\ell=0}^{\infty} (\ell+\frac{1}{2})\, d_\ell P_\ell(\mu) \sum_{j=0}^{\infty} (j+\frac{1}{2})\, f_j(x, s) \int_{-1}^{1} d\mu'\, P_\ell(\mu')\, P_j(\mu') = \\
&\frac{2\pi}{\lambda} \sum_{\ell=0}^{\infty} (\ell+\frac{1}{2})\, d_\ell P_\ell(\mu) \sum_{j=0}^{\infty} (j+\frac{1}{2})\, f_j(x, s) \left[\frac{2}{2j+1}\, \delta_{\ell j}\right] = \\
&\frac{2\pi}{\lambda} \sum_{j=0}^{\infty} \frac{2j+1}{2}\, d_j\, f_j(x, s)\, P_j(\mu) \quad .
\end{aligned} \tag{15}$$

The following identity is useful for dealing with the $\mu \frac{\partial f}{\partial x}$ term of the left-hand side of Eqn. (6).

$$\mu\, P_\ell(\mu) = \frac{(\ell+1)\, P_{\ell+1}(\mu) + \ell\, P_{\ell-1}(\mu)}{2\ell+1} \quad . \tag{16}$$

Then the gradient term becomes

$$\mu \frac{\partial f}{\partial x}(x,\mu,s) = \frac{1}{2}\sum_{\ell=0}^{\infty}\Big[(\ell+1)\,P_{\ell+1}(\mu) + \ell\,P_{\ell-1}(\mu)\Big]\frac{\partial f_{\ell}}{\partial x}(x,s) \quad . \tag{17}$$

If $\ell+1=j$ in the first sum and $\ell-1=j$ in the second sum above, then

$$\sum_{\ell=0}^{\infty}(\ell+1)\,P_{\ell+1}(\mu) = \sum_{j=0}^{\infty} j\,P_j(\mu)$$

and

$$\sum_{\ell=0}^{\infty}\ell\,P_{\ell-1}(\mu) = \sum_{j=0}^{\infty}(j+1)\,P_j(\mu) \quad .$$

The left-hand side of Eqn. (6) then becomes

$$\sum_{j=0}^{\infty}\left[(j+\tfrac{1}{2})\,P_j(\mu)\right]\left[\frac{\partial}{\partial s} + \frac{1}{\lambda}\right] f_j(x,s) \; +$$
$$\frac{1}{2}\sum_{j=0}^{\infty}\left[j\,\frac{\partial f_{j-1}}{\partial x}(x,s) + (j+1)\,\frac{\partial f_j}{\partial x}(x,s)\right]P_j(\mu) \quad .$$

When the equal j terms are matched on both sides of Eqn. (6) there results the following set of j+1 coupled partial differential equations for the Legendre coefficients of the electron flux:

$$\left(\frac{\partial}{\partial s} + \frac{1}{\lambda}\right) f_j(x,s) + \left(\frac{j}{2j+1}\right)\frac{\partial f_{j-1}}{\partial x}(x,s) + \left(\frac{j+1}{2j+1}\right)\frac{\partial f_{j+1}}{\partial x}(x,s) =$$
$$\frac{2\pi}{\lambda}d_j f_j(x,s) + S_j(x,s) \quad , \tag{18}$$
$$j=0,1,2.... \quad .$$

The above equations represent a simplification of the problem in that the angular variable has been removed, and the flux function f(x, μ, s) has been replaced by a set of angle-independent flux, expansion coefficients $f_j(x, s)$. These equations can be solved numerically using finite difference methods. It must be pointed out that there are at least three important factors to be considered in the implementation of finite difference solutions of the above equations: 1) the spatial (x) behavior of the flux; 2) the rapidity of flux variation with pathlength(s); 3) flux shape anisotropy; 4) degree of angular anisotropy of the scattering. The first two factors directly influence the calculation in an obvious way. The x and s grid sizes must be fine enough to track rapid or abrupt variations in the spatial or pathlength dependence of the flux coefficients. Also, a "rule of thumb" for finite difference calculations must be factored in; the spatial integration step size ought not to exceed λ/10. These requirements can place strenuous demands on both the core storage and computation time consumed by such a calculation, particularly if the flux distribution is derived from a monodirectional point source. The third and, fourth factors, flux shape and scattering anisotropies, influence the feasibility of the calculation through the number of Legendre series terms required to render an adequate representation of the flux angular distribution (Eqn. 8) in the former case, and the number of Legendre coefficients d_j required to accurately describe the scattering process in the latter. These two factors are not entirely separable since flux anisotropy can result from either or both of two causes, an initial monodirectional flux distribution and anisotropic scattering. The net effect is that of increasing the number of coupled equations (18) to be solved.

2.2.2 Slab Boundary Conditions

The calculation of the Legendre coefficients of the electron flux is performed for slabs of finite thickness. This fact arises both out of computational necessity, the limitation due to the number of spatial integration steps, and an attempt to accurately model the physical situation. Marshak[(5)] boundary conditions were adopted to describe the behavior of the Legendre flux coefficients at the slab extremeties. The Marshak conditions are vacuum boundary conditions on the particle current. Given a slab of width 'a' (Fig. 2-1), the requirement of zero return current from the vacuum may be expressed in terms of the following integral conditions:

$$\left.\begin{array}{l} \displaystyle\int_{o}^{1} f(o,\mu,s)\, P_m(\mu)d\mu \;=\; 0 \\[2ex] \text{and} \\ \displaystyle\int_{-1}^{o} f(a,\mu,s)\, P_m(\mu)d\mu \;=\; 0 \end{array}\right\} \; m=1,3,5,\ldots \qquad (19)$$

The Legendre coefficients at the slab boundaries, $f_\ell(o,s)$ and $f_\ell(a,s)$ were solved for directly using the above conditions (19) in conjunction with Eqn. (8) to form the following set of linear equations.

$$\left.\begin{array}{l} \displaystyle\sum_{\ell} (\ell+\tfrac{1}{2}) f_\ell(o,s) \int_{o}^{1} P_\ell(\mu)\, P_m(\mu)\; d\mu \;=\; 0 \\[3ex] \displaystyle\sum_{\ell} (\ell+\tfrac{1}{2}) f_\ell(a,s) \int_{-1}^{o} P_\ell(\mu)\, P_m(\mu)\; d\mu \;=\; 0 \end{array}\right\} \; \begin{array}{l} m=1,3,5,7\ldots \\ \ell=0,1,2,3\ldots \end{array} \qquad (20)$$

The coefficients $f_\ell(o,s)$ and $f_\ell(a,s)$ were computed for ℓ values ranging from 0 to 20.

2.2.3 Calculation of Legendre Expansion Coefficients for Screened-Rutherford Scattering

An alternative to the expression for the screened-Rutherford scattering probability density given by Eqn. (4) is an expression in terms of the scattering deflection angle α

$$P(\cos\alpha) = \frac{1}{2\pi}\,\frac{\eta(1+\eta/2)}{(1+\eta-\cos\alpha)^2}\ . \tag{21}$$

The expansion coefficients, q_ℓ, of Eqn. (10) are obtained by multiplying both sides of Eqn. (10) by $P_j(\cos\alpha)$ and integrating. That is

$$\begin{aligned}&\int_{-1}^{1} P_j(\cos\alpha)\,P(\cos\alpha)\,d(\cos\alpha) = \\ &\sum_{\ell=0}^{\infty}(\ell+\tfrac{1}{2})\,d_\ell\int_{-1}^{1} P_\ell(\cos\alpha)\,P_j(\cos\alpha)\,d(\cos\alpha) = d_\ell\,\delta_{\ell j}\ ,\end{aligned} \tag{22}$$

or

$$\begin{aligned}d_\ell &= \int_{-1}^{1} P_\ell(\cos\alpha)\,P(\cos\alpha)\,d(\cos\alpha) \\ &= \frac{1}{2\pi}\,\eta(1+\frac{\eta}{2})\int_{-1}^{1}\frac{P_\ell(\cos\alpha)\,d(\cos\alpha)}{(1+\eta-\cos\alpha)^2}\ .\end{aligned} \tag{23}$$

Garth[6] has derived a set of relations for obtaining the d_ℓ's which eliminates the necessity for evaluating the integrals of Eqn. (23). If a set of quantities $C_\ell(\eta)$ are defined as

$$C_\ell(\eta) = d_o(\eta) - d_\ell(\eta) \tag{24}$$

where
$$d_o(\eta) = \frac{1}{\eta(1+\eta/2)}\ , \tag{25}$$

then the following recursion relation holds for the C_ℓ's:

$$\ell\, C_{\ell+1} = (2\ell+1)\,(1+\eta)\, C_\ell \; - \; (\ell+1)\, C_{\ell-1} \; - \; \frac{2\ell+1}{1+\frac{\eta}{2}} \quad . \tag{26}$$

The above relations were used to obtain the screened-Rutherford scattering expansion coefficients in the P_ℓ computer code.

2.2.4 Numerical Integration of Coupled Equations for Legendre Coefficients

Two finite difference schemes were employed to integrate the set of coupled equations for the Legendre coefficients of the flux, $f_j(x, s)$. The simpler of the two is the Friedrichs(7) scheme, an explicit method, in which the derivatives of f are estimated linearly as would seem appropriate for solving a linear hyperbolic equation. The other integration scheme is a second order calculation which utilizes the coupling between the equations for different order Legendre coefficients to obtain more accurate estimates of the derivatives of f. The results obtained with the second order scheme are more accurate than those given by the Friedrichs method with little noticeable sacrifice in computational cost.

2.2.4.1 The Friedrichs Scheme

Numerical integration of the system of equations (18) by means of the Friedrichs scheme utilizes forward differencing along the path-length variable, s, and a divided difference derivative formula along the spatial variable, x. With reference to the x, s grid of Figure 2-2, the system of equations (18) can be re-written in finite difference form according to the Friedrichs scheme as follows:

$$\frac{(f_\ell)_k^{n+1} - \frac{1}{2}\left[(f_\ell)_{k+1}^{n} + (f_\ell)_{k-1}^{n}\right]}{\Delta s} + \alpha_\ell \, \frac{\left(f_{\ell-1}\right)_{k+1}^{n} - \left(f_{\ell-1}\right)_{k-1}^{n}}{2\Delta x}$$

$$+\beta_\ell \, \frac{\left(f_{\ell+1}\right)_{k+1}^{n} - \left(f_{\ell+1}\right)_{k-1}^{n}}{2\Delta x} + \frac{1}{2\lambda}\left[\left(f_\ell\right)_{k+1}^{n} + \left(f_\ell\right)_{k-1}^{n}\right] = \qquad (27)$$

$$\frac{d_\ell}{2\lambda}\left[\left(f_\ell\right)_{k+1}^{n} + \left(f_\ell\right)_{k-1}^{n}\right] \quad ; \quad \ell=0,1,2\ldots,\ell\text{max}.$$

where $\alpha_\ell = \frac{\ell}{2\ell+1}$, $\beta = \frac{\ell+1}{2\ell+1}$,

ℓmax is the index of the highest order turn to be evaluated in the Legendre series, and Δs and Δx are the finite intervals corresponding respectively to the differentials ds and dx. The quantity $\left(f_\ell\right)_k^{n+1}$ is the value of the ℓ-th order Legendre flux coefficient at the x, s grid point P. Analogous definitions apply to the other bracketed quantities of Eqn. (27). The essential difference between this scheme and a simple explicit scheme is that the value of $f_\ell(x, s)$ at the grid point R, namely $\left(f_\ell\right)_k^{n}$, is replaced everywhere by the average over the values at the two adjacent x grid points, $\frac{1}{2}\left[\left(f_\ell\right)_{k+1}^{n} + \left(f_\ell\right)_{k-1}^{n}\right]$. Conditions necessary for the implementation of this scheme are;

$$\left.\begin{matrix} f_{-1}(x, s) \\ \\ f_{\ell\text{max}+1}(x, s) \end{matrix}\right\} \equiv 0 \quad \text{for all } x, s. \qquad (28)$$

Then if either a source condition or, as is usually the case, an initial condition on the flux coefficients $f_{\ell}(x, o)$ for all ℓ and x values is given, the values of the Legendre coefficients of the flux at the grid point P can be obtained. If the quantities A and B are defined as

$$A \equiv \frac{\Delta s}{2\Delta x} \quad , \tag{29a}$$

and

$$B \equiv \frac{\Delta s}{\lambda} \quad , \tag{29b}$$

then

$$\begin{aligned}(f_{\ell})^{n+1}_{k} = {} & \frac{1}{2}\left\{1 + B(q_{\ell}-1)\right\}\left\{(f_{\ell})^{n}_{k+1} + (f_{\ell})^{n}_{k-1}\right\} \\ & + A\left\{\alpha_{\ell}\left(f_{\ell-1}\right)^{n}_{k-1} + \beta_{\ell}\left(f_{\ell+1}\right)^{n}_{k-1}\right\} \\ & - A\left\{\alpha_{\ell}\left(f_{\ell-1}\right)^{n}_{k+1} + \beta_{\ell}\left(f_{\ell+1}\right)^{n}_{k+1}\right\} + \lambda B\left(S_{\ell}\right)^{n}_{k} \quad .\end{aligned} \tag{30}$$

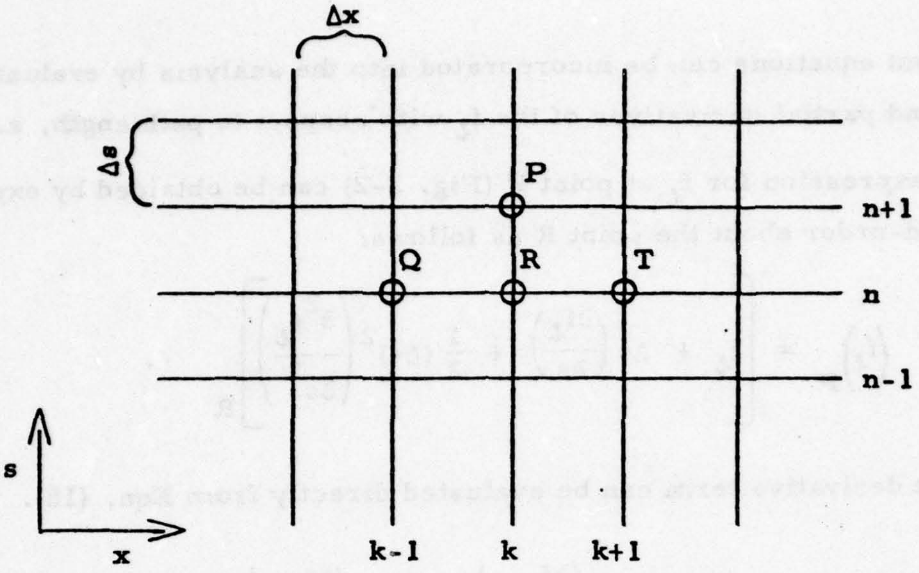


Fig. 2-2.

The (x, s) grid

The computational procedure employed to implement Eqn. (30) consists essentially of the execution of three nested loops. The sequence of calculation is: a) evaluate the Legendre coefficients at a particular point (x, s) for all values of ℓ, $0 \le \ell \le \ell\text{max}$ (innermost loop); b) perform the above (a) for every x value on the (x, s) grid at a fixed s value (middle loop); c) perform the above (a∩b) for every s value on the (x, s) grid.

2.2.4.2 The Second-Order Integration Scheme

The second-order integration scheme, an adaptation of the Lax-Wendroff[(8)] method, yields an improvement in accuracy over the spatial averaging scheme of Friedrichs. The applicability of a second order scheme to the system of coupled first-order linear equations (18) arises from the fact that the coupling among the various order Legendre coefficient equations can be incorporated into the analysis by evaluating the second partial derivatives of the f_ℓ with respect to pathlength, s.

An expression for f_ℓ at point P (Fig. 2-2) can be obtained by expansion to second-order about the point R as follows:

$$\left(f_\ell\right)_P = \left[f_\ell + \Delta s \left(\frac{\partial f_\ell}{\partial s}\right) + \frac{1}{2}(\Delta s)^2 \left(\frac{\partial^2 f_\ell}{\partial s^2}\right)\right]_R \quad . \tag{31}$$

The first derivative term can be evaluated directly from Eqn. (18).

$$\left(\frac{\partial f_\ell}{\partial s}\right)_R = \frac{1}{\lambda}(d_\ell - 1)\left(f_\ell\right)_R - \alpha_\ell \left(\frac{\partial f_{\ell-1}}{\partial x}\right)_R - \beta_\ell \left(\frac{\partial f_{\ell+1}}{\partial x}\right)_R + \left(S_\ell\right)_R \quad . \tag{32}$$

Differentiation of the above expression (32) once with respect to s yields an expression for the second partial derivative of Eqn. (31). That is

$$\left(\frac{\partial^2 f_\ell}{\partial s^2}\right)_R = \frac{1}{\lambda}(d_\ell - 1)\left(\frac{\partial f_\ell}{\partial s}\right)_R - \alpha_\ell \left(\frac{\partial^2 f_{\ell-1}}{\partial s \partial x}\right)_R - \beta_\ell \left(\frac{\partial^2 f_{\ell+1}}{\partial s \partial x}\right)_R + \left(\frac{\partial S_\ell}{\partial s}\right)_R \quad . \tag{33}$$

Upon substitution of Eqn. (32) into Eqn. (33), the second derivative becomes

$$\begin{aligned}\left(\frac{\partial^2 f_\ell}{\partial s^2}\right)_R &= \frac{1}{\lambda^2}(d_\ell - 1)^2 \left(f_\ell\right)_R - \frac{\alpha_\ell}{\lambda}(d_\ell - 1)\left(\frac{\partial f_{\ell-1}}{\partial x}\right)_R - \frac{\beta_\ell}{\lambda}(d_\ell - 1)\left(\frac{\partial f_{\ell+1}}{\partial x}\right)_R \\ &+ \frac{1}{\lambda}(d_\ell - 1)\left(S_\ell\right)_R - \alpha_\ell \left(\frac{\partial^2 f_{\ell-1}}{\partial s \partial x}\right)_R - \beta_\ell \left(\frac{\partial^2 f_{\ell+1}}{\partial s \partial \alpha}\right)_R + \left(\frac{\partial S_\ell}{\partial s}\right)_R \quad .\end{aligned} \tag{34}$$

The mixed partial derivatives can be evaluated in terms of spatial derivatives alone by rewriting Eqn. (18) for order ℓ-1;

$$\left[\frac{\partial}{\partial s} - \frac{1}{\lambda}(d_{\ell-1} -1)\right] f_{\ell-1} + \alpha_{\ell-1}\frac{\partial f_{\ell-2}}{\partial x} + \beta_{\ell-1}\frac{\partial f_{\ell}}{\partial x} - S_{\ell-1} = 0 \quad , \tag{35}$$

for order ℓ+1;

$$\left[\frac{\partial}{\partial s} - \frac{1}{\lambda}(d_{\ell+1} -1)\right] f_{\ell+1} + \alpha_{\ell+1}\frac{\partial f_{\ell}}{\partial x} + \beta_{\ell+1}\frac{\partial f_{\ell+2}}{\partial x} - S_{\ell+1} = 0 \quad , \tag{36}$$

and then differentiating each expression (35, 36) once with respect to x, yielding

$$\frac{\partial^2 f_{\ell-1}}{\partial s \partial x} = \frac{1}{\lambda}(d_{\ell-1}-1)\frac{\partial f_{\ell-1}}{\partial x} - \alpha_{\ell-1}\frac{\partial^2 f_{\ell-2}}{\partial x^2} - \beta_{\ell-1}\frac{\partial^2 f_{\ell}}{\partial x^2} + \frac{\partial S_{\ell-1}}{\partial x} \tag{37}$$

and

$$\frac{\partial^2 f_{\ell+1}}{\partial s \partial x} = \frac{1}{\lambda}(d_{\ell+1}-1)\frac{\partial f_{\ell+1}}{\partial x} - \alpha_{\ell+1}\frac{\partial^2 f_{\ell}}{\partial x^2} - \beta_{\ell+1}\frac{\partial^2 f_{\ell+2}}{\partial x^2} + \frac{\partial S_{\ell+1}}{\partial x} \quad . \tag{38}$$

If Eqns. (37) and (38) are substituted into Eqn. (34), and if this result is then used along with Eqn. (32) to eliminate the derivatives of f with respect to s in Eqn. (31), the final expression for f_ℓ at point P is

$$
\begin{aligned}
(f_\ell)_P = & \left\{1 + \frac{\Delta s}{\lambda}(d_\ell - 1) + \frac{1}{2}\left(\frac{\Delta s}{\lambda}\right)^2 (d_\ell - 1)^2\right\}(f_\ell)_R \\
& - \left\{\Delta s + \frac{1}{2}\frac{(\Delta s)^2}{\lambda}(d_\ell + d_{\ell-1} - 2)\right\}\alpha_\ell\left(\frac{\partial f_{\ell-1}}{\partial x}\right)_R \\
& - \left\{\Delta s + \frac{1}{2}\frac{(\Delta s)^2}{\lambda}(d_\ell + d_{\ell+1} - 2)\right\}\beta_\ell\left(\frac{\partial f_{\ell+1}}{\partial x}\right)_R \qquad (39) \\
& + \frac{1}{2}(\Delta s)^2\left\{\alpha_\ell\alpha_{\ell-1}\left(\frac{\partial^2 f_{\ell-2}}{\partial x^2}\right)_R + \beta_\ell\beta_{\ell+1}\left(\frac{\partial^2 f_{\ell+2}}{\partial x^2}\right)_R + \left(\alpha_\ell\beta_{\ell-1} + \beta_\ell\alpha_{\ell+1}\right)\left(\frac{\partial^2 f_\ell}{\partial x^2}\right)_R\right\} \\
& + \frac{1}{2}(\Delta s)^2\left\{\left(\frac{\partial S_\ell}{\partial s}\right)_R - \alpha_\ell\left(\frac{\partial S_{\ell-1}}{\partial x}\right)_R - \beta_\ell\left(\frac{\partial S_{\ell+1}}{\partial x}\right)_R + \frac{(d_\ell - 1)}{\lambda}S_{\ell_R}\right\} + S_{\ell_R}\Delta s \quad .
\end{aligned}
$$

From Eqn. (39) it is seen that exploitation of the coupling feature of the equations for the Legendre flux coefficients yields a second order expression for the values of the Legendre coefficients f_ℓ at grid point P are obtainable in terms of the f_ℓ and their spatial derivatives only at point R. If source terms are present, their spatial and pathlength derivatives can be evaluated either analytically or numerically.

2.2.4.3 Comparison of Friedrichs and Second-Order Scheme Results

Numerical results were obtained using both the second-order integration scheme and the Friedrichs scheme for a test case where comparison with exact flux values was readily obtainable, the P_1 (ℓmax = 1) equations in the absence of scattering (streaming). Rather than a source term, S, a Gaussian spatially distributed monodirectional (direction cosine μ_o) initial flux was assumed. Table 2.1 is a comparison of the numerical results for the total flux f_o obtained with the two methods with the exact results at the approximate peak flux point, $x \approx x_o + \mu_o s$, in a slab of unit thickness. The finite difference steps in x and s were both set equal to 0.01 slab widths, and λ was taken to be 10^{10} slab widths (virtually no scattering).

Table 2.1

Comparison of Total Flux, f_o, obtained with Second-Order and Friedrichs Scheme with Exact Values for Peak Flux Points, $\Delta x = \Delta x = 0.01$

s (slab widths)	Second-Order Scheme	Friedrichs Scheme	Exact Values
0.1	5.637	5.299	5.639
0.2	5.624	5.010	5.630
0.3	5.635	5.009	5.636
0.4	5.641	4.554	5.641
0.5	5.627	4.370	5.641
0.6	5.612	4.205	5.634

Also, a comparison of the second-order scheme results with the exact values when the numerical integration was performed along the characteristic line $\Delta s = \Delta x/\mu_o$ yielded discrepancies in the fourth significant figure only.

2.2.5 Orders-of-Scattering Flux Calculations

The Spencer-Lewis equation (1) can be expressed in terms of a set of equations for the electron flux if the flux is expanded in an orders-of-scattering series.

Let

$$f(x, \mu, s) = \sum_{n=0}^{\infty} f_n(x, \mu, s) \tag{40}$$

where n is the number of elastic scatterings undergone by the electron along its trajectory in the scattering medium. Substitution of the above series into Eqn. (1) leads to a set of n+1 orders-of-scattering equations:

$$\begin{aligned} \frac{\partial f_o}{\partial s} + \mu \frac{\partial f_o}{\partial x} + \frac{f_o}{\lambda(s)} &= S(x, \mu, s) \\ &\vdots \\ \frac{\partial f_n}{\partial s} + \mu \frac{\partial f_n}{\partial x} + \frac{f_n}{\lambda(s)} &= S_n(x, \mu, s) \end{aligned} \tag{41}$$

where

$$S_n(x, \mu, s) = \frac{1}{\lambda(s)} \int_{-1}^{1} d\mu' \; P(\mu' \rightarrow \mu, s) \, f_{n-1}(x, \mu', s) \quad . \tag{42}$$

The functions S_n can be considered to be sources of n-times scattered particles or collision densities.

Correspondingly, numerical solutions can be derived for the Legendre coefficients of the n-th scattered electron flux. In direct analogy to equations (18), the coupled set of equations for the n-th scattered flux coefficients is

$$\left(\frac{\partial}{\partial s} + \frac{1}{\lambda}\right) f_j^n(x, s) + \left(\frac{j}{2j+1}\right)\left(\frac{\partial f_{j-1}^n(x, s)}{\partial x}\right) + \left(\frac{j+1}{2j+1}\right)\left(\frac{\partial f_{j+1}^n(x, s)}{\partial x}\right) = S_j^n(x, s) \quad ; \quad j=0, 1, 2.... \tag{43}$$

where f_j^n and S_j^n are respectively the j-th Legendre coefficients of the n-th scattered flux and source function. The finite difference solution, to second order, for the $f_j^n(x, s)$ is of the same form as that given for the Legendre flux coefficients in Eqn. (39). Aside from affixing a superscript n to the appropriate quantities, the only remaining modification consists of setting the scattering coefficients d_j to zero. A version of the P_ℓ method code was prepared to execute this algorithm.

2.2.6 Physical Quantities Computed by the P_ℓ Method Code (PLMETHD)

The computer code (PLMETHD) written to implement the method of solution described in the preceding sections (2.2 et seq.) utilizes the Legendre coefficients of the electron flux to calculate, in subroutine PROFILE, the following set of physical quantities:

1) forward charge flow profile
2) backward charge flow profile
3) net charge flow profile
4) forward energy flow profile
5) backward energy flow profile
6) net energy flow profile
7) energy deposition profile

The mathematical definitions of these quantities are given in Ref. (1).

A listing of the code PLMETHD, the version which utilizes the second-order integration scheme (sec. 2.2.4.2), is given in Appendix 1.

2.3 Solution of the Transport Equation by the Method of Characteristics

2.3.1 General Discussion

The Spencer-Lewis equation for the flux of n-times scattered electrons,

$$\frac{\partial f_n}{\partial s} + \mu \frac{\partial f_n}{\partial x} + \frac{f_n}{\lambda} = S_n(x, \mu, s) \quad , \tag{41}$$

can be solved using the method of characteristics,[9] an outline of which will now be given.

If the substitution $f_n(x, \mu, s) = e^{-s/\lambda} h_n(x, \mu, s)$ is used in Eqn. (41), the resulting solution will be

$$h_n(x, \mu, s) = \int_{s_1}^{s} e^{t/\lambda} S_n[x - \mu(s-t), \mu, t]\, dt \quad . \tag{44}$$

In the above, t is the dummy variable of integration, and s_1 is the lower limit of integration (which will be determined later). The first argument of the function S_n in the integrand, $x - \mu(s-t)$ is the projection on the x-axis of the particle path between collisions and, in fact, represents the x position of the particle which has already traversed a total pathlength t (Fig. 2-3), and having arrived at the site of the n-th collision, will have traversed a total pathlength s.

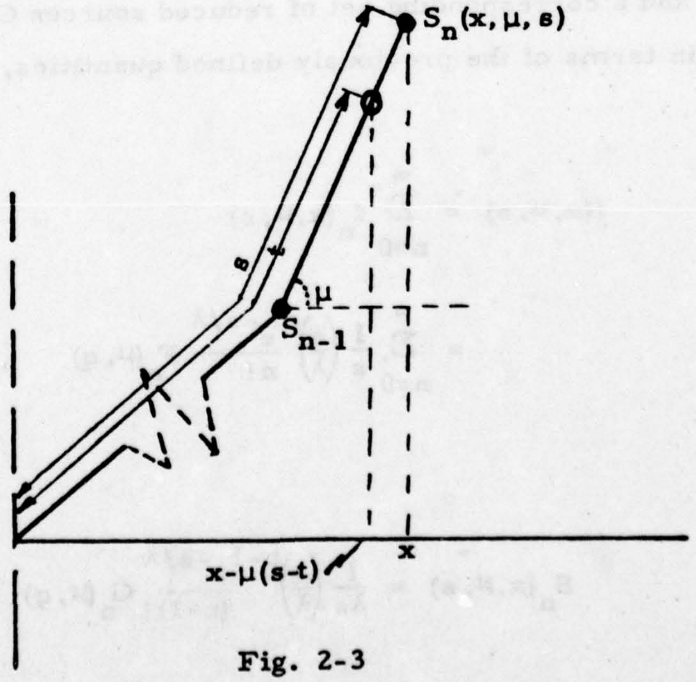


Fig. 2-3

Finally, upon conversion of the h_n functions back to the original flux functions, the solution of Eqn. (41) by the method of characteristics is

$$f_n(x,\mu,s) = e^{-s/\lambda} \int_{s_1}^{s} e^{t/\lambda} S_n(x-\mu s+\mu t,\mu,t)\, dt \tag{45}$$

where $$S_n(x,\mu,s) = \frac{1}{\lambda} \int_{-1}^{1} d\mu' \, P(\mu'\to\mu)\, f_{n-1}(x,\mu',s) \quad . \tag{46}$$

2.3.2 Expression of Solution in terms of Reduced Source and Flux Functions

While the above integral expressions (45, 46) provide a solution of the orders-of-scattering equations (41) in the formal sense, these are not yet cast in the form most amenable to numerical solution. The work of Ganapol et al(10, 11) has suggested a formulation of the flux and source functions which has proven to be a valuable aid in this regard. A simplification of the problem is achieved through the expression of the flux and source functions in terms the quotient $q = x/s$ and μ, a reduction of the number of variables from three to two.

For the n-th order-of-scattering, a set of reduced flux functions $F_n(\mu, q)$ and a corresponding set of reduced sources $G_n(\mu, q)$ can be defined in terms of the previously defined quantities, f, f_n and S_n as follows:

$$\begin{aligned} f(x,\mu,s) &= \sum_{n=0}^{\infty} f_n(x,\mu,s) \\ &= \sum_{n=0}^{\infty} \frac{1}{s}\left(\frac{s}{\lambda}\right)^n \frac{e^{-s/\lambda}}{n!} F_n(\mu,q) \quad , \end{aligned} \tag{47}$$

and

$$S_n(x,\mu,s) = \frac{1}{\lambda s}\left(\frac{s}{\lambda}\right)^{n-1} \frac{e^{-s/\lambda}}{(n-1)!} G_n(\mu,q) \tag{48}$$

where $q \equiv x/s$.

An additional simplifying assumption, that the scattering probability P is independent of s, was made.

2.3.2.1 Recursion Relations Between Reduced Flux and Source Functions

If use is made of the definitions, Eqns. (47, 48), the following recursion relation between the reduced functions F_n and G_n may be derived from Eqn. (46). The resulting relations are

$$G_n(\mu, q) = \int_{-1}^{1} d\mu' \, P(\mu' \rightarrow \mu) \, F_{n-1}(\mu', q) \tag{49}$$

and

$$F_n(\mu, q) = n \int_{\rho_1}^{1} d\rho \, \rho^{n-2} \, G_n(\mu, q') \quad , \tag{50}$$

where

$$q' = \frac{q-\mu}{\rho} + \mu \quad . \tag{51}$$

The lower limit of the above integral has not been specified. The specification of ρ_1 and correspondingly, s_1 (Eqn. 45) can be accomplished by considering the possible relationships between the variables μ and q. The overriding constraints on the quantities of Eqn. (65) are

$$\left\{ \begin{array}{c} 0 \le \rho \le 1 \\ -1 \le \mu \le 1 \\ -1 \le q \le 1 \\ -1 \le q' \le 1 \end{array} \right\} \tag{52}$$

If all of the possible relations between q and μ, consistent with the above constraints, are considered, the resulting values for ρ_1 are:

$$\rho_1 = \frac{q-\mu}{1-\mu} \quad , \quad (q>\mu) \quad ; \tag{53}$$

$$\rho_1 = \frac{\mu-q}{1-\mu} \quad , \quad (q<\mu) \quad ; \tag{54}$$

$$\rho_1 = 0 \quad , \quad (q=\mu) \quad . \tag{55}$$

Correspondingly, for s, the results are:

$$s_1 = \frac{x-\mu s}{1-\mu} \quad , \quad (x>\mu s) \quad ; \tag{56}$$

$$s_1 = \frac{\mu s-x}{1+\mu} \quad , \quad (x<\mu s) \quad ; \tag{57}$$

$$s_1 = 0 \quad , \quad (x=\mu s) \tag{58}$$

The relations given by Eqns. (49) and (50), while containing one less variable than their counterparts, Eqns. (45) and (46), do not yet constitute a basis for numerical evaluation of the F_n and G_n functions. Implementation of Eqns. (49) and (50) is complicated by the necessity for evaluating two separate integrals, one for $q<\mu$ and another for $q>\mu$. Furthermore, for $n<2$, the integrand of Eqn. (50) is singular at the point $q = \mu$. The recursion equations (49) and (50) can, however, be employed to deriving a tractable recursion formula for the G_n functions.

2.3.2.2 A Practical Recursion Relation for the G_n Functions

The interrelationships between the F_n and G_n given Eqns. (49) and (50) can be combined to yield the following integral equation for the reduced source functions:

$$G_{n+1}(\mu, q) = n \int_{-1}^{q} dt \int_{q}^{1} dr \left[\frac{P(t,\mu)\,(q-t)^{n-1} G_n(t,r) + P(r,\mu)\,(r-q)^{n-1} G_n(r,t)}{(r-t)^n} \right] . \tag{59}$$

At the point where the integrand denominator goes to zero (r=t=q), the numerator is also equal to zero. This presents no real difficulty since the integrand is well behaved on either side of r=t=q. The numerical integration can be safely executed about this point without evaluating the integrand at it.

This final form of the G_n recursion relationship provides a practical means for obtaining the reduced source functions for the n-th order of scattering. The only requirement for implementation of this formula is that an adequate numerical or analytical representation of a G_n surface to initiate the iterative procedure be supplied by another means for some low value of n. Thus far, numerical calculations involving Eqn. (59) have been performed for a monodirectional point source, with the $G_2(\mu, q)$ surface supplying the initial integrand values, and for a point isotropic source, with the $G_1(\mu, q)$ surface playing the same role. Both screened-Rutherford and isotropic scattering were considered. The case of isotropic scatter of particles emanating from a monodirectional source normally directed with respect to the slab surface will be discussed first since it affords some qualitative insight into the behavior of the G_n functions as n increases with a comparatively small computational effort.

2.3.3 Monodirectional (Normal) Source; Isotropic Scattering

The application of Eqn. (59) to the special case of isotropic scatter from a monodirectional point source can prove useful in investigating the behavior of reduced source functions as the order of scattering increases. The recursion relation reduces to a simpler analytical form for this case, since the scattering probability densities, $P(r,\mu)$ and $P(t,\mu)$, take on constant values $\left(= \frac{1}{2}\right)$. Not only is the writing of a computer program to evaluate Eqn. (59) a relatively simple task, but it is possible, as will be shown, to obtain an independent analytical evaluation of $G_3(q)$ for this case. Thus a benchmark curve is available with which to check the numerical integration program.

A monodirectional source of unscattered particles at the (x=0) in an infinite medium is represented mathematically as

$$S_o(x, \mu, s) = \delta(x)\,\delta(1-\mu)\,\delta(s) \quad . \tag{60}$$

Substitution of this expression into Eqn. (54) yields for the unscattered flux

$$f_o(x, \mu, s) = e^{-s/\lambda}\,\delta(x-s)\,\delta(1-\mu) \quad . \tag{61}$$

The source of once-scattered particles and the once-scattered particle flux may be obtained by the successive application of Eqns. (46) and (45), taking $P(\mu' \rightarrow \mu) = \frac{1}{2}$. That is

or

$$S_1(x, \mu, s) = \frac{e^{-s/\lambda}}{2\lambda}\,\delta(x-s) \quad , \tag{62}$$

and

$$f_1(x, \mu, s) = \begin{cases} \dfrac{e^{-s/\lambda}}{2\lambda}\left[\dfrac{\Theta(x-\mu s) - \Theta(x-s)}{1-\mu}\right] \quad , & |x| < s \\ 0 & |x| > s \end{cases} \tag{63}$$

where $$\Theta(z) = \begin{Bmatrix} 1 \quad , & z \leq 0 \\ 0 \quad , & z > 0 \end{Bmatrix} \quad , \tag{64}$$

is the Heaviside step function.

Before the conversion to reduced source and flux functions, this process can be repeated for another step. Again from Eqn. (46), one has

$$S_2(x,\mu,s) = \frac{e^{-s/\lambda}}{4\lambda^2} \, \ell n \left(\frac{2s}{s-x}\right) \quad , \tag{65}$$

and from Eqn. (45),

$$f_2(x,\mu,s) = \frac{e^{-s/\lambda}}{4\lambda^2} \begin{cases} \Big[s \, \ell n \, 2s - \left(\frac{x-\mu s}{1-\mu}\right) \ell n \left(\frac{2(x-\mu s)}{1-\mu}\right) - \left(\frac{s-x}{1-\mu}\right) \ell n \, (s-x) \ , & x > \mu s \\ s \, \ell n \left(\frac{2}{1-\mu}\right) \quad , & x = \mu s \\ \Big[s \, \ell n \, 2s + \left(\frac{\mu s-x}{1-\mu}\right) \ell n \left(\frac{2(\mu s-x)}{1+\mu}\right) - \left(\frac{s-x}{1-\mu}\right) \ell n \, (s-x) \ , & x < \mu s \\ 0 \quad , & s = 0 \end{cases} \tag{66}$$

where the lower limits of integration, s_1, in Eqn. (45), are supplied by Eqns. (56, 57, 58).

Expressions for the corresponding reduced source and flux functions can be written immediately using the definitions of Eqns. (47) and (48).

$$F_o(\mu, q) = \delta(q-1) \, \delta(1-\mu) \quad , \tag{67}$$

$$F_1(\mu, q) = \frac{\Theta(q-\mu) - \Theta(q-1)}{2(1-\mu)} \quad , \tag{68}$$

$$F_2(\mu, q) = \begin{cases} \frac{1}{2}\left[\left(\frac{1-q}{1-\mu}\right) \ell n \left(\frac{2}{1-q}\right) + \left(\frac{q-\mu}{1-\mu}\right) \ell n \left(\frac{1-\mu}{q-\mu}\right)\right] \quad , & q > \mu \\ \frac{1}{2} \ell n \left(\frac{2}{1-q}\right) \quad , & q = \mu \\ \frac{1}{2}\left[\left(\frac{1-q}{1-\mu}\right) \ell n \left(\frac{2}{1-q}\right) + \left(\frac{\mu-q}{1-\mu}\right) \ell n \left(\frac{\mu-q}{1+\mu}\right)\right] \quad , & q < \mu \quad , \end{cases} \tag{69}$$

$$G_1(\mu, q) = \frac{1}{2} \delta(q-1) \quad , \tag{70}$$

$$G_2(\mu, q) = \frac{1}{4} \ell n \left(\frac{2}{1-q}\right) \quad . \tag{71}$$

From the above and Eqn. (49), it is immediately observable that for isotropic scattering the G_n functions are independent of μ. This fact permits an analytic simplification for the recursion relation of the G_n. If the functions g_n are defined as

$$g_n(q) \equiv G_n(\mu, q) \quad , \tag{72}$$

then under the assumption of isotropic scattering, Eqn. (59) may be rewritten as

$$g_{n+1}(q) = \frac{n}{2}\left[\int_q^1 dq'\, g_n(q')\, I_n(q,q') + \int_{-1}^q dq'\, g_n(q')\, J_n(q,q')\right] . \tag{73}$$

where for notational convenience, the integrals over μ are defined as

$$I_n(q,q') \equiv \int_{-1}^q d\mu\, \frac{(q-\mu)^{n-1}}{(q'-\mu)^n} , \quad J_n(q,q') = \int_q^1 d\mu\, \frac{(\mu-q)^{n-1}}{(\mu-q')^n} , \tag{74}$$

The functions I_n and J_n can be evaluated recursively. For n=1 and n=2, the integrals of Eqn. (74) are found in standard integral tables(12).

A computer code, GAUSQN, was written to evaluate the $g_n(q)$ to order n=10. The iterations were begun with $g_2(q)$ as given by Eqns. (71, 72). Figure 2-4 is a plot of the $g_n(q)$ curves for n=3-10. The q' integrations of Eqn. (73) were performed numerically using 16 point Gaussian quadrature. This choice of quadrature eliminated the necessity to evaluate the integrand g_2 at the singular point q=1. A code listing of GAUSQN is given in Appendix 2.

In order to assess the magnitude of numerical error introduced as a consequence of the selection of a quadrature technique which avoided the endpoint singularity of g_2, an effort was made to evaluate the function $g_3(q)$ analytically. A direct analytical integration of Eqn. (73) was performed with the expressions for I_2 and J_2 as given by Eqn. (74). The result of this effort yielded the following benchmark expression for $g_3(q)$:

$$g_3(q) = \frac{(1-q)}{2}\left[\ln\left(\frac{1-q}{1+q}\right)\ln\left(\frac{1-q}{2}\right) - 2\ln(1-q) + L_{i2}\left(\frac{1-q}{2}\right) - L_{i2}\left(\frac{1+q}{2}\right) - \frac{\pi^2}{6}\right] \tag{75}$$

$$+ 2\ln 2 - (1+q)\ln(1+q) ,$$

where $L_{i2}(x)$ is the dilogarithm function and is given by[13,14]

$$L_{i2}(x) \equiv - \int_0^x \frac{\ln(1-x)}{x}\, dx = \sum_1^\infty \frac{x^n}{n^2} \quad . \tag{76}$$

Numerical values of $g_3(q)$ obtained via the two methods, numerical integration and evaluation of the above analytical expression, were found to be in substantial agreement, as shown in Table 2.2. Thus the validity of the numerical integration scheme was established.

2.3.4 Monodirectional (Normal) Source; Screened-Rutherford Scattering

Unlike the isotropic scattering case, arrival at an analytic solution of Eqn. (74) is highly improbable for screened-Rutherford scattering. Two independent numerical solution methods for obtaining the G_n functions for $n \geq 3$ were devised. The first of these is a computationally expensive "brute force" numerical integration of Eqn. (74), as it reads, using Gaussian quadrature in conjunction with a surface interpolation algorithm to evaluate the integrand G_n functions at the Gaussian ordinates. The second method is based on the expansion of the G_n functions in a Legendre series in μ, and solving a set of integral equations for the Legendre coefficients of G_n. This second scheme has proven to be considerably more economical than the first when the scattering anisotropy is mild enough to justify truncation of the Legendre series to about 10 or 15 terms. In addition to the Legendre expansion of the angular dependence of the G_n functions, a further economy is achieved through the approximation of the q dependent portion of G_n by Chebyshev series. A certain amount of discretion in determining the appropriate use of these approximations is acquired through experience, so that as investigations proceed from one screened-Rutherford case (η value) to another the process of evaluating the G_n functions gains in efficiency.

As will be demonstrated in a later section, the validity of the two numerical methods used to integrate Eqn. (74) was independently confirmed by the application of the P_ℓ method to the integration of the Spencer-Lewis equation with the orders-of-scattering source provided by the G_n functions.

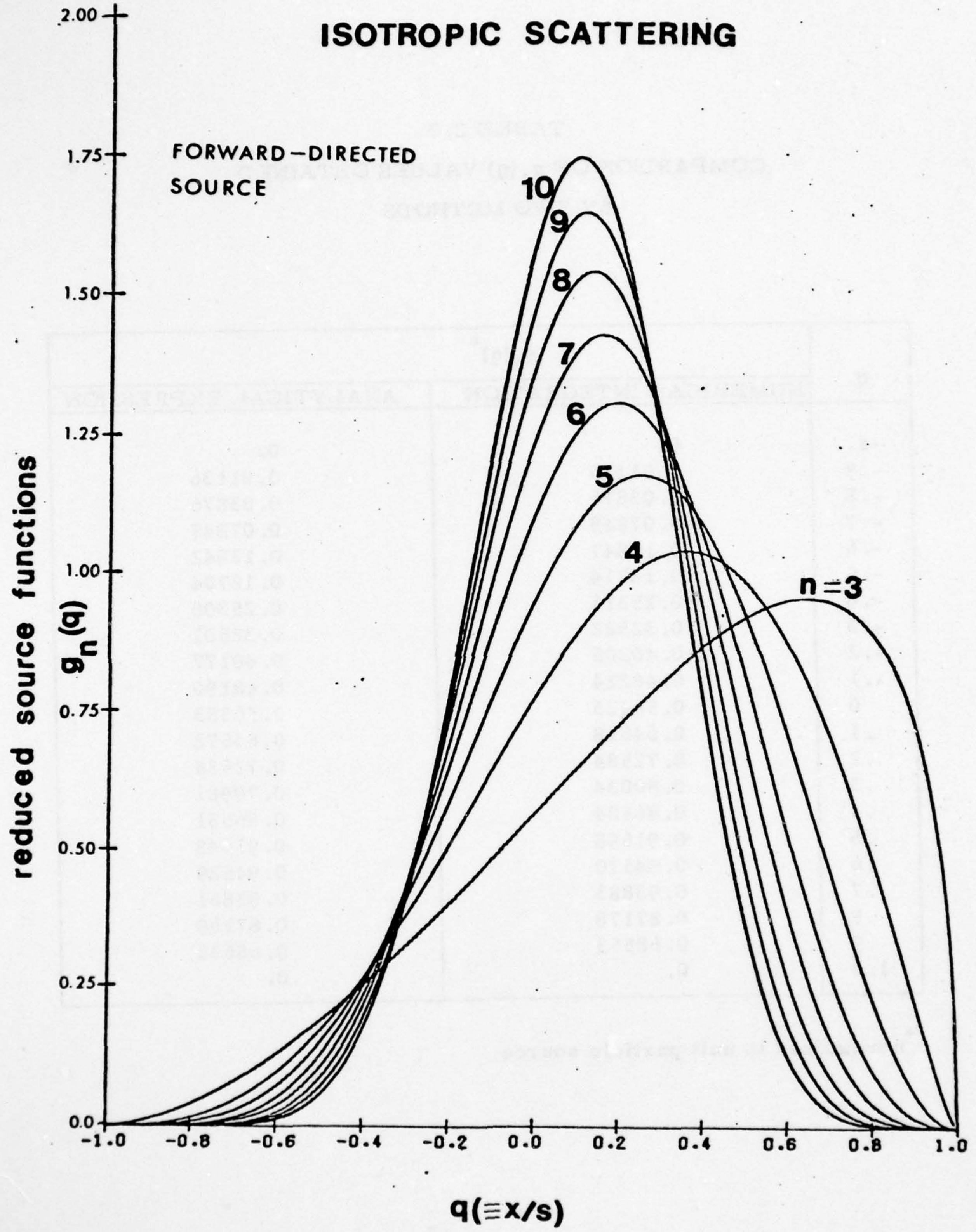

ISOTROPIC SCATTERING
FORWARD-DIRECTED
SOURCE
10
9
8
7
6
5
4
n=3
reduced source functions $g_n(q)$
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.0
-1.0
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
$q(\equiv x/s)$

Fig. 2-4

TABLE 2.2

COMPARISON OF $g_3(q)$ VALUES OBTAINED BY TWO METHODS

q	$g_3(q)^*$	
	NUMERICAL INTEGRATION	ANALYTICAL EXPRESSION
-1.	0.	0.
-.9	0.01135	0.01136
-.8	0.03876	0.03876
-.7	0.07849	0.07847
-.6	0.12847	0.12842
-.5	0.18714	0.18704
-.4	0.25316	0.25300
-.3	0.32522	0.32501
-.2	0.40205	0.40177
-.1	0.48224	0.48190
0	0.56423	0.56383
.1	0.64618	0.64572
.2	0.72584	0.72534
.3	0.80034	0.79981
.4	0.86584	0.86531
.5	0.91698	0.91648
.6	0.94570	0.94529
.7	0.93883	0.93861
.8	0.87178	0.87189
.9	0.68553	0.68632
1.0	0.	0.

*normalized to unit particle source

2.3.4.1 Recursive Algorithm Initiation

Independent of the choice of either of the above two integration methods is the problem of obtaining a set of $G_n(\mu, q)$ values with which the recursion relations of Eqn. (59) may be initialized. An analytical expression for $G_n(\mu, q)$ can be obtained for n=2. This provides a departure point from which the numerical integration algorithm can be applied.

As was done for the isotropic scattering case, successive application of Eqns. (45) and (46), to the extent that analytical evaluation of the integrals is feasible, results in an expression for the source function $S_2(x, \mu, s)$ from which the $G_2(\mu, q)$ function may in turn be obtained via Eqn. (48). The applicable scattering probability density here is, of course, the screened-Rutherford probability, given by Eqn. (4), which is a function of both the initial and final direction cosines as contrasted with the constant value of 0.5 for isotropic scatter. If the procedure used to obtain Eqns. (61-71) after a considerable amount of algebraic manipulation, the following expression for G_2:

$$G_2(\mu, q) = \left[\frac{y(1+\frac{\eta}{2})}{\eta}\right]^2 \left[\begin{array}{l} -\ (1+3y+\frac{4}{\eta})\, I_1 \ + \ (3y+3+\frac{6}{\eta})\, I_2 \\ -\ (y+3+\frac{4}{\eta})\, I_3 \ + \ (1+\frac{1}{\eta})\, I_4 \ + \ (y+\frac{1}{\eta})\, I_5 \end{array}\right] , \tag{77}$$

where

$$\left.\begin{array}{l}
I_1 = \dfrac{-2(2\gamma+\beta x)}{(4\alpha\gamma-\beta^2)\,(\alpha x^2+\beta x+\gamma)^{1/2}}\Bigg|_{x_1}^{x_2} , \\[2ex]
I_2 = \dfrac{2(2\alpha x+\beta)}{(4\alpha\gamma-\beta^2)\,(\alpha x^2+\beta x+\gamma)^{1/2}}\Bigg|_{x_1}^{x_2} , \\[2ex]
I_3 = \\
\qquad - \dfrac{2(\alpha\beta x-2\alpha\gamma+\beta^2)}{\gamma(4\alpha\gamma-\beta^2)\,(\alpha x^2+\beta x+\gamma)^{1/2}} - \dfrac{1}{\gamma^{3/2}}\, \ell n \left[\dfrac{2\gamma+\beta x+2\sqrt{\gamma(\alpha x^2+\beta x+\gamma)}}{x}\right]\Bigg|_{x_1}^{x_2} , \\[2ex]
I_4 = \quad = -\dfrac{1}{\gamma x(\alpha x^2+\beta x+\gamma)^{1/2}}\Bigg|_{x_1}^{x_2} - \dfrac{3\beta}{2\gamma} I_3 - \dfrac{2\alpha}{\gamma} I_2 , \\[2ex]
I_5 = \quad = \\
\qquad \dfrac{-(4\alpha\gamma-2\beta^2)x-2\gamma\beta}{\alpha(4\alpha\gamma-\beta^2)\,(\alpha x^2+\beta x+\gamma)^{1/2}} \quad \dfrac{1}{\alpha^{3/2}}\, \ell n \left[2\sqrt{\alpha(\alpha x^2+\beta x+\gamma)} \ + \ 2\alpha x+\beta\right]\Bigg|_{x_1}^{x_2} ,
\end{array}\right\} \tag{78}$$

and $$\alpha = \left(y + \frac{1}{\eta}\right)^2 \quad ,$$

$$\beta = -2\left(\frac{3y}{\eta} + y + \frac{1}{\eta^2}\right) \quad ,$$

$$\gamma = 2y\left(1 + \frac{2}{\eta}\right) + \frac{1}{\eta^2} \quad ,$$

$$x_1 = 1 + {}^{\eta}/_{2} \quad ,$$

$$x_2 = 1 + \frac{\eta}{1-q} \quad ,$$

$$y = \frac{1}{1-\mu} \quad .$$

A program was written to evaluate the above expression (Eqn. (77) for $G_2(\mu, q)$. This program has been incorporated as a subroutine in the numerical integration program.

2.3.4.2 Numerical Integration of the G_n Recursion Equation via the Gaussian Quadrature Technique

A numerical integration of Eqn. (59), and hence the evaluation of the $G_n(\mu, q)$ functions for various values of the Rutherford screening parameter, was performed using the Gaussian quadrature technique. If the Gaussian ordinates for a J-th order quadrature calculation are denoted as ξ_j, $(1 \le j \le J)$ (the ξ_j are the zeros of the Legendre polynomial of order J), and the corresponding Gaussian weighting coefficients as A_j, then Eqn. (59) when restated in the discrete ordinate representation appears as

$$G_{n+1}(\mu_k, q_\ell) = \sum_{j=1}^{J} A_j \sum_{i=1}^{J} A_i \left[\frac{P(t_j^\ell, \mu_k)\,(q_\ell - t_j^\ell)^{n-1}\,G_n(t_j^\ell, r_i^\ell) + P(r_i^\ell, \mu_k)\,(r_i^\ell - q_\ell)\,G_n(r_i^\ell, t_j^\ell)}{(r_i^\ell - t_j^\ell)^n}\right] \quad , \qquad (78)$$

where the function G_{n+1} is evaluated at the point $\left(\mu_k, q_\ell\right)$ and

$$r_i^{\ell} = \frac{1}{2}\left[(1+q_{\ell}) + (1-q_{\ell})\,\xi_i\right] \quad , \tag{79}$$

$$t_j^{\ell} = \frac{1}{2}\left[(q_{\ell}-1) + (1+q_{\ell})\,\xi_j\right] \quad . \tag{80}$$

A program (ANISO) was written to perform the calculation of Eqn. (78). A listing of ANISO is supplied in Appendix 2. The function G_{n+1} was evaluated for 201 values of μ_k and 201 values of q_{ℓ}, both spanning the interval (-1, 1). Most commonly, the μ and q arrays were taken as equispaced. However, when the range of large G_{n+1} or rapidly varying values was found to be highly concentrated in a small region of the (μ, q) grid, the spacing of the μ and q arrays was adjusted to produce an appropriately high resolution in the rapidly varying region of the G_{n+1} surface. The calculation was initiated with the $G_2(\mu, q)$ computed from the analytical expression of Eqn. (77) at all the points on the (201 x 201) (μ, q) grid. Evaluation of G_n at the Gaussian ordinate points (r_i^{ℓ}, t_j^{ℓ}), (t_j^{ℓ}, r_i^{ℓ}) was accomplished by means of a surface interpolation subroutine (SURFTRP) based on Prenter's[15] algorithm in which cubic Hermite polynomials are employed as the interpolation weighting functions.

Calculations based on Eqn. (78) as described above proved to be rather costly to operate. A 32 point Gaussian quadrature was used to insure sufficient accuracy. The accuracy could be monitored in part by integrating G_n over all μ and q values and comparing the result with unity (the G_n were assumed normalized to unit particle incidence). Given a grid of 201 x 201 (μ, q) points, and a 32 point quadrature, execution of the calculation consumed a total of $201 \times 201 \times 32 \times 32 \approx 4.1 \times 10^7$ evaluations of the integrand of Eqn. (78) for each order of scattering. In terms of CDC 6600 central processor time, a complete determination of $G_n(\mu, q)$ required 1500 secs. per scattering order. Due to the expense of these calculations, the ANISO code was exercised only for the lowest scattering orders (G_3 in all cases and G_4 in cases of extreme anisotropy). As will become evident in the graphs of $G_n(q)$ (Figs. 2-6, 2-11), the behavior of the G_n functions for screened-Rutherford scattering was observed to approach that found in the isotropic scattering case as the scattering order is increased. For this reason it was felt that use of a more approximate method for solving Eqn. (59) could be justified for the higher scattering orders ($n \geq 4$). An algorithm based for a more economical calculation, on expansion of the G_n functions in a Legendre series was developed and implemented.

2.3.4.3 Numerical Integration of the G_n Recursion Equation via the Legendre Series Expansion Method

The form of the integral recursion relation for the G_n functions which lends itself more readily to the Legendre series expansion method of solutions is given by Eqn. (58) where identification of the angular variables, μ and μ', is made. The later version of this expression, Eqn. (59) while simpler in form, sacrifice some physical significance through the use of dummy variables to achieve mathematical simplification. The Legendre series method is generally applicable when the series expansions are performed in terms of angular variables.

With reference to Eqn. (73), let the Legendre expansions for G_n and P be written as

$$G_n(\mu', q') = \sum_{\ell=0}^{\infty} (\ell + \tfrac{1}{2})\, g_{n,\ell}(q')\, P_\ell(\mu') \quad , \qquad (81)$$

and

$$P(\mu' \to \mu) = \sum_{\ell=0}^{\infty} (\ell + \tfrac{1}{2})\, d_\ell\, P_\ell(\mu)\, P_\ell(\mu') \quad , \qquad (82)$$

where $g_{n,\ell}(q')$ are the Legendre expansion coefficients of G_n, d_ℓ are the Legendre expansion coefficients and the P_ℓ are Legendre polynomials of order ℓ.

If the expression for $G_n(\mu', q')$ given by Eqn. (81) is substituted into the right hand side of Eqn. (58), then after some algebraic manipulation a recursion formula for the g's is obtained.

This approach to the solution of the reduced source function recursion equation has the advantage that the summation of the Legendre series can be truncated at the point where the coefficients, d_ℓ, of the scattering kernel become small enough to render significant contributions to the result. No such adaptability is afforded by the previously described calculational procedure based on Eqn. (78). Another advantage of evaluating Legendre coefficients of G_n is that the relative magnitudes of the $g_{n,\ell}$ as ℓ is varied, are a direct measure of the degree of anisotropy of the G_n functions. Although this feature has not been fully exploited, it would be possible to devise an adaptive algorithm that utilizes the magnitudes of the $g_{n,\ell}$ to monitor and alter the Legendre series order of the calculation as the order of scattering, n, is increased.

Numerical evaluation of the Legendre coefficients, $g_{n,\ell}(q)$, was accomplished using the method of Gaussian quadrature. As in the case of the direct $G_n(\mu, q)$ calculation (Eqn. (78)), if the Gaussian ordinates and weighting coefficients are respectively defined as ξ_j and A_j, then the expression for the $g_{n,\ell}$ in discrete ordinate representation has the form

$$g_{n+1,\ell}(q') = d_\ell \sum_{\ell'=0}^{L} (\ell'+\frac{1}{2}) \quad .$$

$$\left[\sum_{j=1}^{J} A_j \sum_{i=1}^{J} A_i \left\{ \begin{array}{l} g_{n,\ell'}(q_j)\, P_\ell(\mu'_i)\, P_{\ell'}(\mu'_i) \dfrac{(q'-\mu'_i)^{n-1}}{(q_j-\mu'_i)^n} + \\[2em] g_{n,\ell'}(q_i)\, P_\ell(\mu'_j)\, P_{\ell'}(\mu'_j) \dfrac{(\mu'_j - q')^{n-1}}{(\mu'_j - q_i)^n} \end{array} \right\} \right] , \qquad (83)$$

where L = maximum order of Legendre series,

J = order of Gaussian quadrature,

and

$$q_j = \mu'_j = \frac{1}{2}\left[(1+q') + (1-q')\,\xi_j\right] \qquad (84)$$

$$q_i = \mu'_i = \frac{1}{2}\left[-(1-q') + (1+q')\,\xi_i\right] \quad . \qquad (85)$$

The values of $g_{n,\ell'}$ at q_i and q_j are obtained by cubic spline interpolation.

A computer program (GNLEG) was written to evaluate the $g_{n,\ell}$ coefficients. Provision was made for varying the order, J, of the quadrature. It was found, based on the normalization criterion, that a 32 point quadrature provided sufficient accuracy when used in conjunction with a Legendre series calculation of sufficiently high order to adequately represent the anisotropy.

Initialization of the GNLEG program is accomplished in one of two ways, depending on the initial order-of-scattering. If the calculation is to begin with an evaluation of the $g_{3,\ell}(q)$ array, the input Legendre coefficients $g_{2,\ell}(q)$ are computed by a call to a subroutine to evaluate $G_2(\mu, q)$ (Eqn. (77) followed by a Gaussian quadrature integration at each of the 201 equispaced q points $(-1 \leq q \leq 1)$ to yield

$$g_{2,\ell}(q) = \sum_{j=1}^{J} A_j\, P_\ell(\xi_j)\, G_2(\xi_j, q) \qquad (86)$$

where the A_j and ξ_j are as previously defined. If, however, the calculation is to be initialized at a higher scattering order, the 201x201 point array $G_n(\mu, q)$ $(n \geq 3)$, previously computed via the ANISO code, is read into GNLEG from a mass storage file, and a 201 point trapezoidal integration is performed over μ at each of 201 values of q as follows

$$g_{n-1,\ell}(q) = \sum_i P_\ell(\mu_i)\, G_{n-1}(\mu_i, q)\Delta\mu \quad , \tag{87}$$

where $\Delta\mu = 0.01$.

The program was written to operate in two modes: 1) evaluation of the $g_{n,\ell}$ at either 51 or 201 equally spaced values of q on the interval (-1, 1); evaluation of the $g_{n,\ell}$ at the 21 or 41 modes, q_c, of the Chebyshev polonomial in q of order 20 or 40 respectively. In the first mode of operation, the 201 points of evaluation coincide exactly with those of the ANISO code, so that $g_{n,0}(q)$ $(= G_n(q))$ corresponds directly to the output of ANISO integrated over μ. The second mode of operation was developed for use at the higher n values where the G_n functions acquire a symmetric character about the point q=0. It was found that an adequate representation of the q dependence of the $g_{n,\ell}$ could be achieved with far fewer than 201 q points by means of a Chebyshev series of order 40 or 20 at higher n values (=8-10). The computational savings introduced by this device provided the means for calculating $G_n(q)$ curves three to ten times faster, depending on the value of the Rutherford screening parameter, than was possible with the ANISO code. Conversion of the $g_{n,\ell}$ data evaluated at the Chebyshev nodes, q_c, to a $g_{n,\ell}$ array evaluated at the 201 equispaced q points was accomplished in the GNLEG code by using the $g_{n,\ell}(q_c)$ values to obtain the coefficients of the Chebyshev series representation of $g_{n,\ell}$, and then evaluating the Chebyshev series at the desired 201 q points. A listing of program GNLEG is given in Appendix 2.

2.3.4.4 Execution of Programs ANISO and GNLEG: Results for Monodirectional Source, Screened-Rutherford Scattering

Several sets of $G_n(\mu, q)$ data were produced as a result of running the programs ANISO and GNLEG in tandem. Because the data generated was voluminous, it is stored in binary file form on magnetic tape. There are three forms of data files. The first consists of the raw output from ANISO, $G_n(\mu, q)$, for 201 values of μ and 201 values of q. The second file contains either Legendre coefficient arrays $g_{n,\ell}(q)$ or Chebyshev-Legendre coefficient arrays. If the number of q points is 51 or greater, the former applies. If not, the file consists of either 21 or 41 $g_{n,\ell}$ coefficients of the Chebyshev series in q. For a given set of n, ℓ values, the length of the stored array is written at the head of the tape file so that the file reading program may interpret the data properly. The third file type results from execution of a program (MAKFIL) which collects the data from both ANISO and GNLEG, by reading the first two file types, and, regardless of origin, transforms it into Legendre coefficients $g_{n,\ell}(q)$, for all orders of scattering, n, all ℓ values, and for 201 values of q, $-1 \leq q \leq 1$. *MAKFIL is a bookkeeping program.* A listing is given in Appendix 2. Another program, LOOK, reads the third file type and prints out the Legendre coefficient arrays $g_{n,\ell}(q)$ in an easily legible form. A listing of LOOK, together with a sample of the printed output, is given in Appendix 2.

Legendre coefficient, $g_{n,\ell}(q)$, data was generated for five values of the Rutherford screening parameter, η(=10., 1., 0.41, 0.2, 0.1) and stored in the manner described above. The computations were done for orders-of-scattering 3 through 10. In the more highly anisotropic cases, the ANISO code was used to generate data for n=3 and n=4, while GNLEG was used to calculate the data for n=5-10. In the remaining three cases ANISO was used for n=3 only. Table 2.3 lists the five η values, their corresponding average scattering direction cosines and angles, the division of computation burden between ANISO and GNLEG, and the order of the GNLEG Legendre series calculation employed. Also listed in the table are the orders-of-scattering for which the various approximation modes of GNLEG are used: 1) Legendre series only; 2) Legendre plus Chebyshev polynomial of order 40; 3) Legendre plus Chebyshev polynomial of order 20. The diagram of Fig. 2-5 is a synopsis of the functions of ANISO and GNLEG and shows the interrelation between them.

For each of the cases listed in Table 2.3, a series of plots were made of the $g_{n,\ell}(q)$ data. A separate CALCOMP plot was made for each order, ℓ, of the Legendre series *coefficient.* Each of these plots displays eight curves, each curve corresponding to an order-of-scattering (n=3-10). The first set of curves $g_{n,0}(q)$ (n=3-10) is exactly equivalent to a plot of the $G_n(q)$ curves since by Eqn. (108)

$$g_{n,0}(q) = \int_{-1}^{1} G_n(\mu, q)\, d\mu \quad .$$

The plots for $\ell \geq 1$ reflect, in a quantitative way, the degree of anisotropy of the scattering and indicate the relative importance, or lack thereof, of the higher order Legendre series terms.

Given are sample results in Figures 2-6 through 2-10 which show the Legendre coefficient curves, $g_{n\ell}(q)$ vs. q, for η=10. Figure 2-6 is a plot of $g_{n,0}(q)$ (=$G_n(q)$), and the remaining four graphs display the higher order (ℓ=1, 2, 3, 4) Legendre coefficient curves. In all of the plots to follow all of the $G_n(q)$ curves are normalized to unit particle incidence.

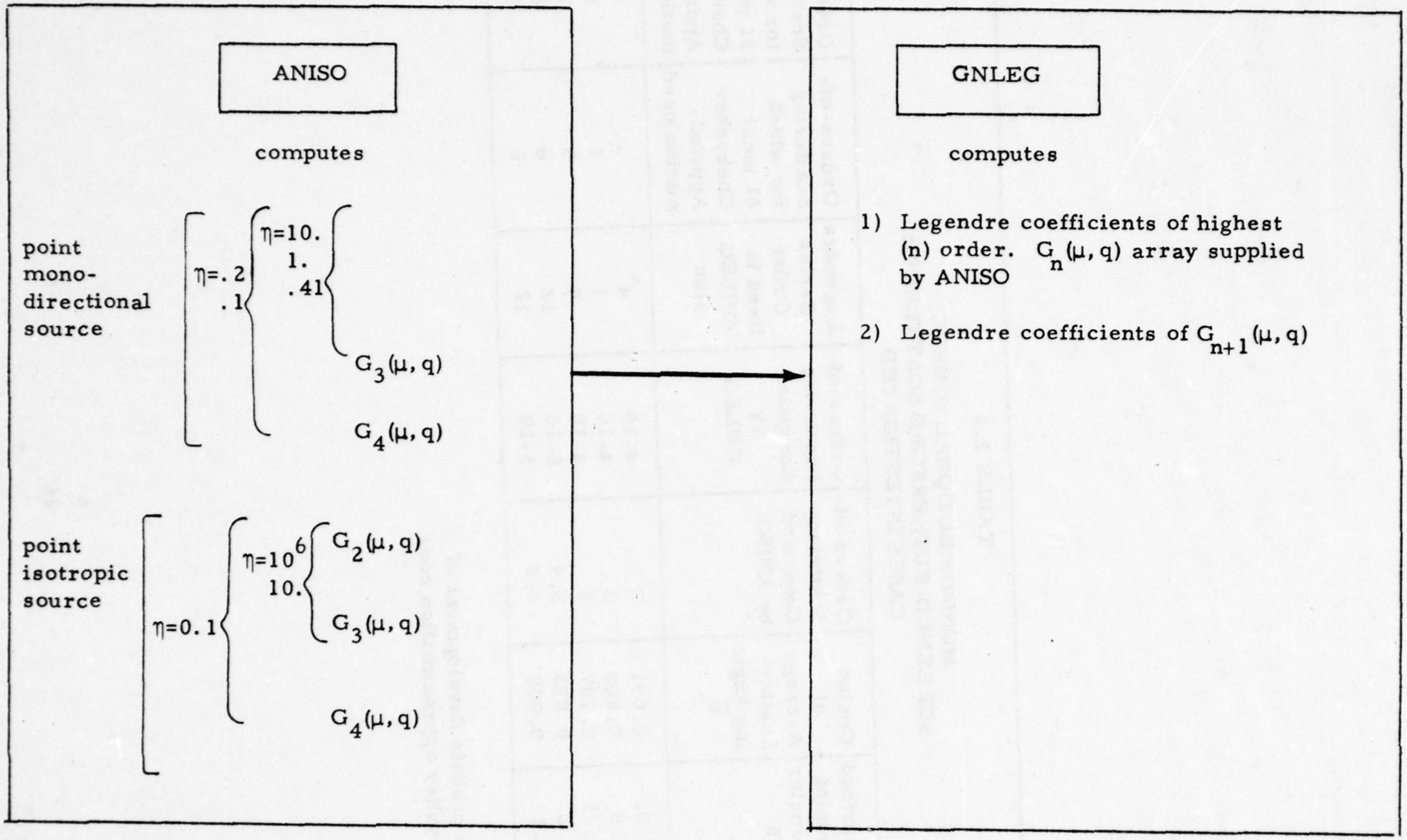

ANISO
computes
point mono-directional source
η=.2
.1
η=10.
1.
.41
$G_3(\mu, q)$
$G_4(\mu, q)$
point isotropic source
η=0.1
$\eta=10^6$
10.
$G_2(\mu, q)$
$G_3(\mu, q)$
$G_4(\mu, q)$
GNLEG
computes
1) Legendre coefficients of highest (n) order. $G_n(\mu, q)$ array supplied by ANISO
2) Legendre coefficients of $G_{n+1}(\mu, q)$

Fig. 2-5

Functions of ANISO and GNLEG Codes

TABLE 2.3

MONODIRECTIONAL SOURCE; SCREENED-RUTHERFORD SCATTERING; CASES INVESTIGATED

Rutherford Screening Parameter η	Cosine of Average Scattering Angle $\bar{\mu}$	Orders-of-Scattering Computed by ANISO	Orders-of-Scattering Computed By GNLEG	Legendre Series Order Used in GNLEG Run	Orders-of-Scattering for which 41 point Chebyshev Approximation used	Orders-of-Scattering for which 21 point Chebyshev Approximation used
10.0	0.091	3	4-10	4*	-	-
1.0	0.500	3	4-10	7	5	6-10
0.41	0.707	3	4-10	9	5	7-10
0.2	0.833	3,4	5-10	12	6	8-10
0.1	0.909	3,4	5-10	12	6	8-10

*Runs predate development of Chebyshev approximation code.

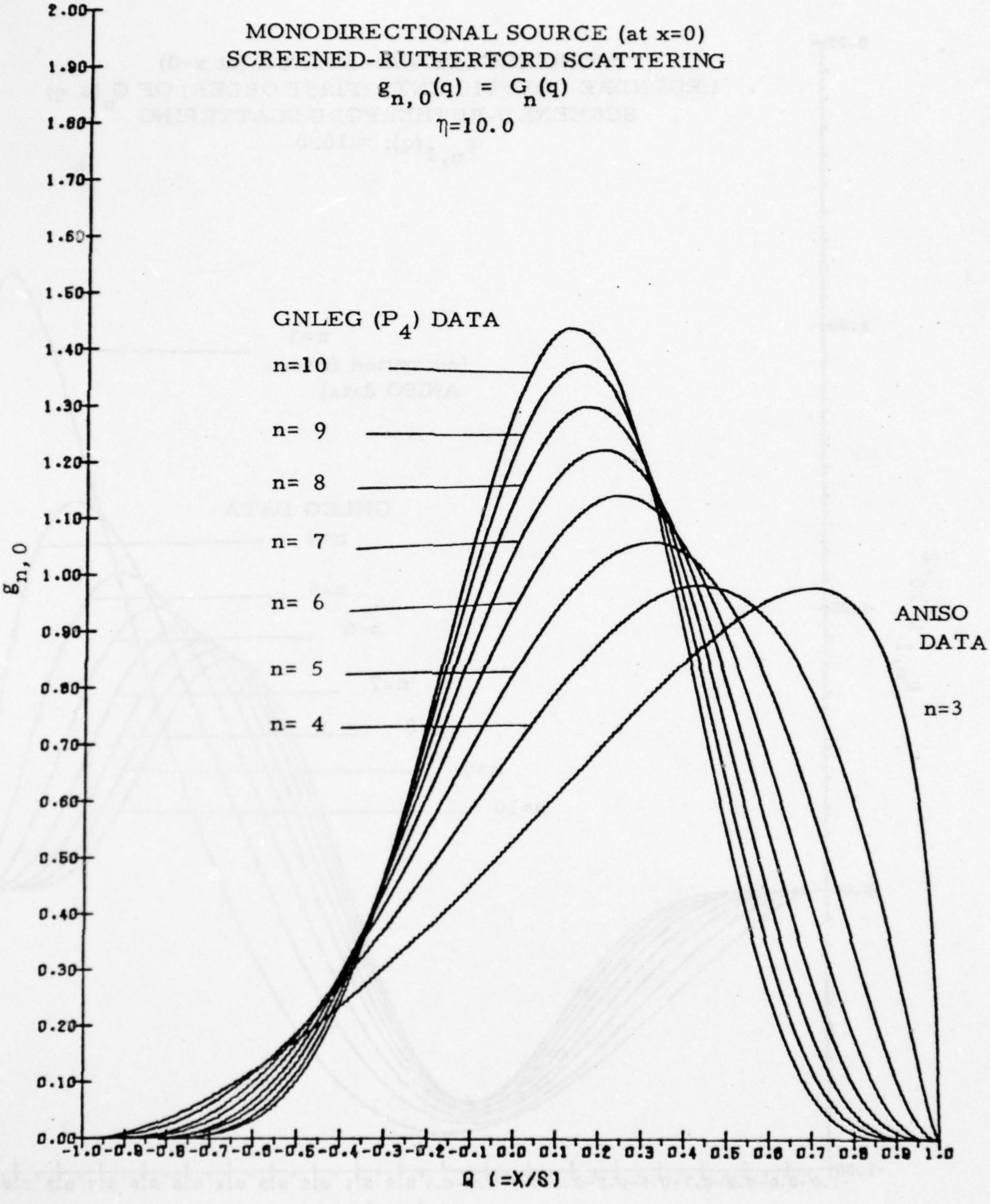
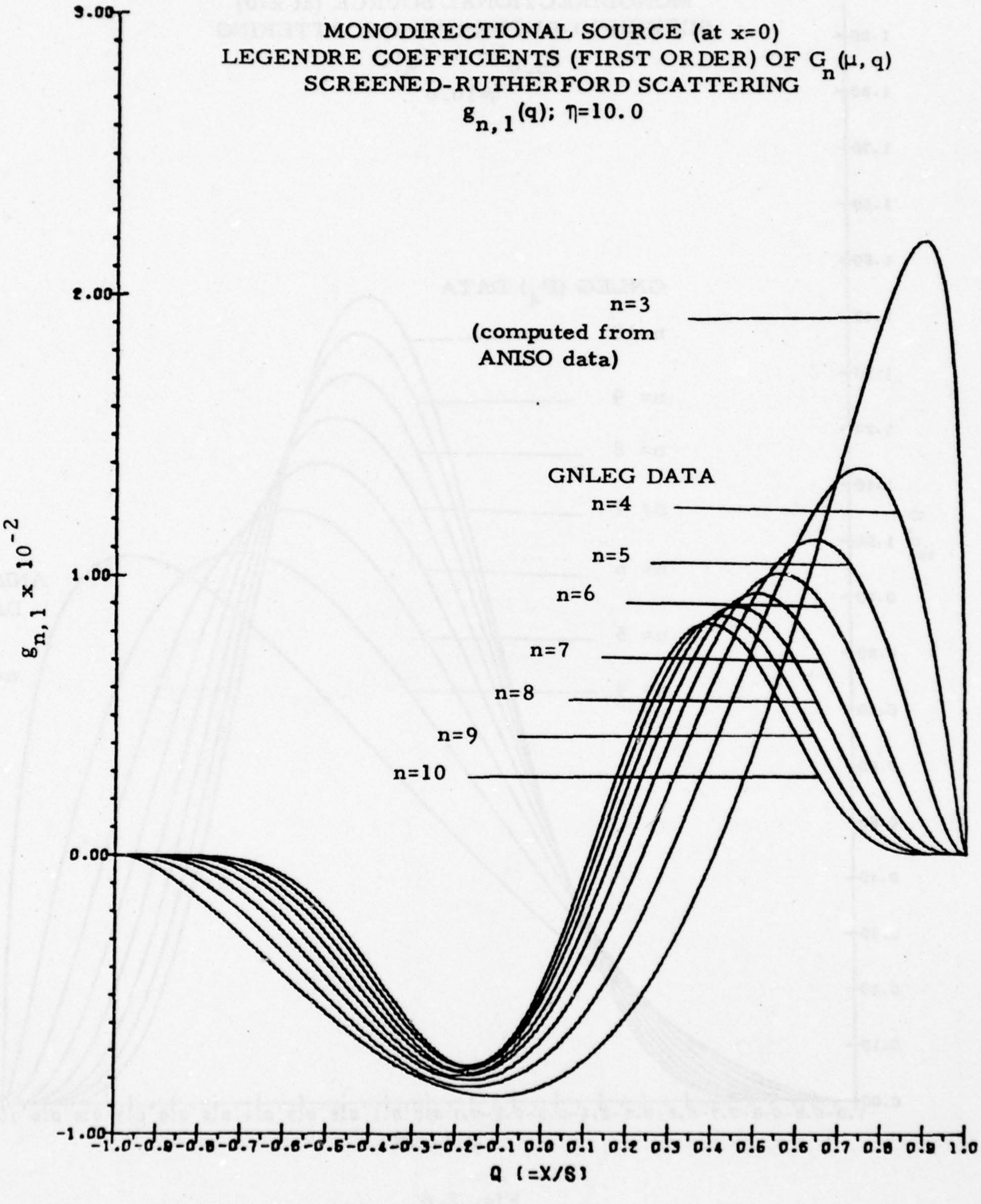

MONODIRECTIONAL SOURCE (at x=0)
SCREENED-RUTHERFORD SCATTERING
$g_{n,0}(q) = G_n(q)$
$\eta=10.0$
GNLEG (P_4) DATA
n=10
n= 9
n= 8
n= 7
n= 6
n= 5
n= 4
ANISO DATA
n=3
$g_{n,0}$
2.00
1.90
1.80
1.70
1.60
1.50
1.40
1.30
1.20
1.10
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
-1.0 -0.9 -0.8 -0.7 -0.6 -0.5 -0.4 -0.3 -0.2 -0.1 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
Q (=X/S)

Fig. 2-6



MONODIRECTIONAL SOURCE (at x=0)
LEGENDRE COEFFICIENTS (FIRST ORDER) OF $G_n(\mu, q)$
SCREENED-RUTHERFORD SCATTERING
$g_{n,1}(q)$; η=10.0

Fig. 2-7

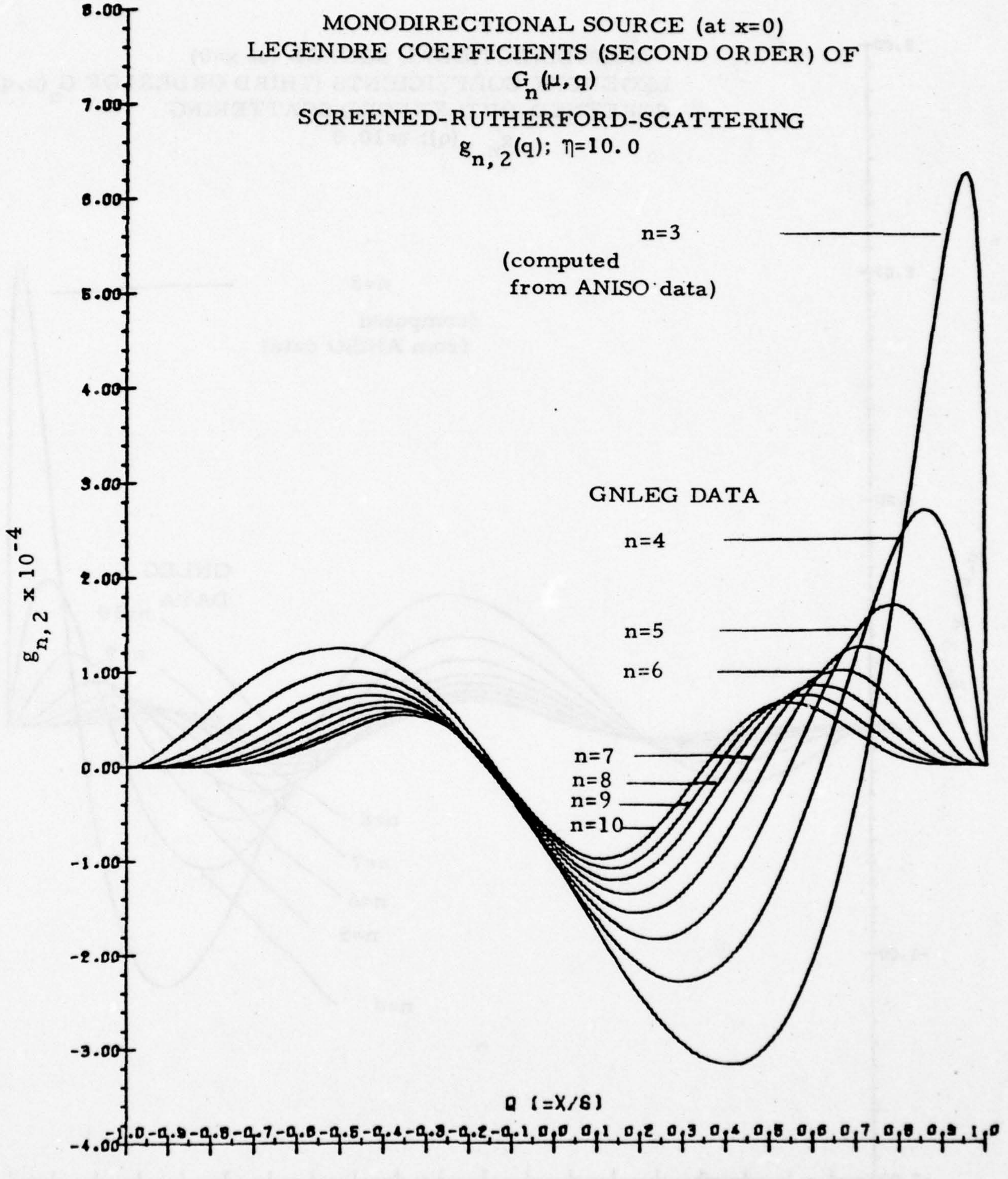

MONODIRECTIONAL SOURCE (at x=0)
LEGENDRE COEFFICIENTS (SECOND ORDER) OF
$G_n(\mu, q)$
SCREENED-RUTHERFORD-SCATTERING
$g_{n,2}(q)$; η=10.0
(computed
from ANISO data)
GNLEG DATA
n=3
n=4
n=5
n=6
n=7
n=8
n=9
n=10
$g_{n,2} \times 10^{-4}$
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
-1.00
-2.00
-3.00
-4.00
Q (=X/S)
-1.0
-0.9
-0.8
-0.7
-0.6
-0.5
-0.4
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

Fig. 2-8

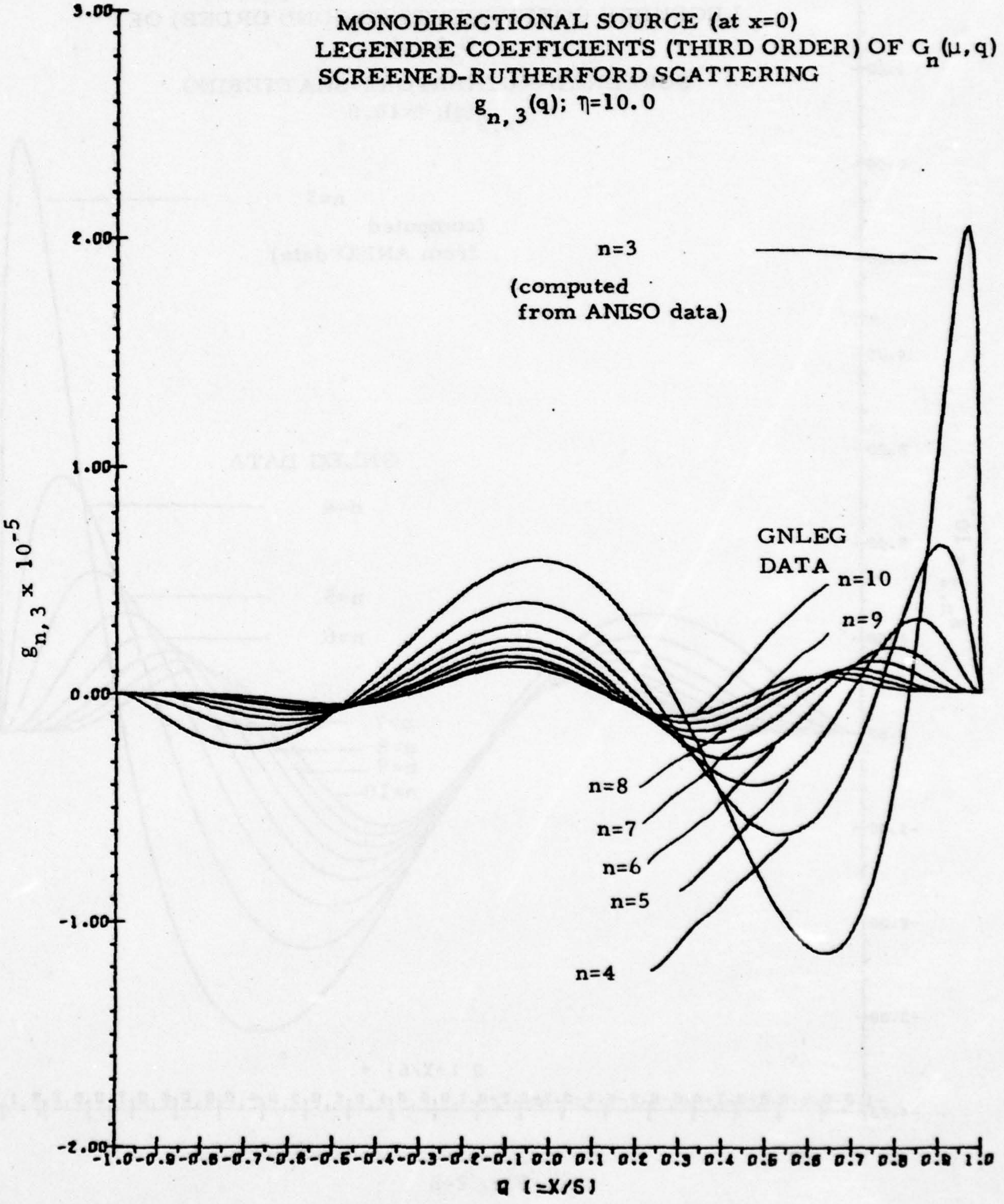

MONODIRECTIONAL SOURCE (at x=0)
LEGENDRE COEFFICIENTS (THIRD ORDER) OF $G_n(\mu, q)$
SCREENED-RUTHERFORD SCATTERING
$g_{n,3}(q)$; η=10.0
n=3
(computed
from ANISO data)
GNLEG
DATA
n=10
n=9
n=8
n=7
n=6
n=5
n=4
$g_{n,3} \times 10^{-5}$
3.00
2.00
1.00
0.00
-1.00
-2.00
-1.0
-0.9
-0.8
-0.7
-0.6
-0.5
-0.4
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Q (=X/S)

Fig. 2-9

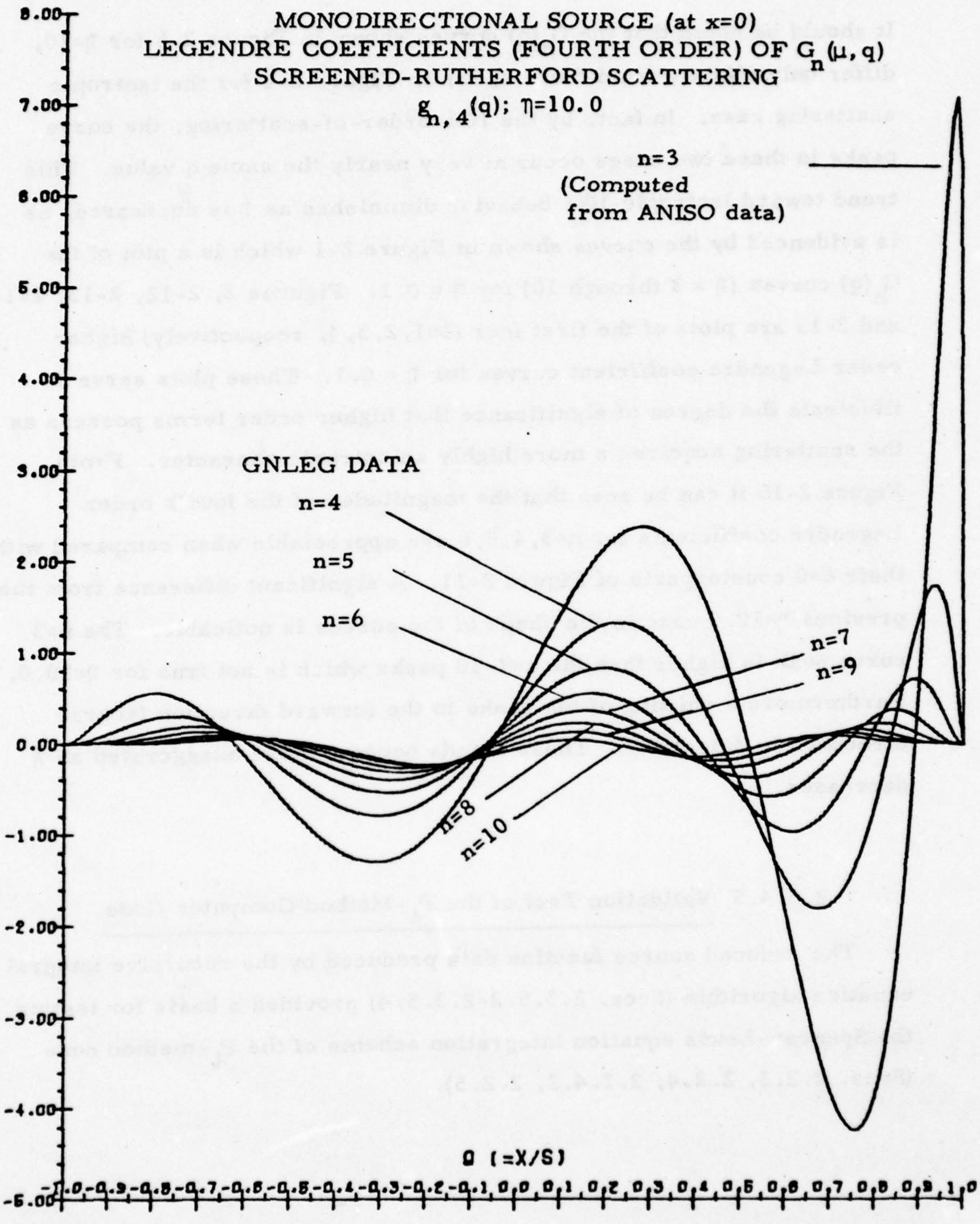

MONODIRECTIONAL SOURCE (at x=0)
LEGENDRE COEFFICIENTS (FOURTH ORDER) OF $G_n(\mu, q)$
SCREENED-RUTHERFORD SCATTERING
$g_{n,4}(q)$; η=10.0
n=3
(Computed
from ANISO data)
GNLEG DATA
n=4
n=5
n=6
n=7
n=9
n=8
n=10
$g_{n,4} \times 10^{-7}$
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
-1.00
-2.00
-3.00
-4.00
-5.00
Q (=X/S)
-1.0
-0.9
-0.8
-0.7
-0.6
-0.5
-0.4
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

Fig. 2-10

It should be noted that the $G_n(q)$ curves shown in Figure 2-6 for η=10, differ only slightly from those shown in Figure 2-4 for the isotropic scattering case. In fact, by the 10th order-of-scattering, the curve peaks in these two cases occur at very nearly the same q value. This trend toward isotropic-like behavior diminishes as η is decreased, as is evidenced by the curves shown in Figure 2-1 which is a plot of the $G_n(q)$ curves (η = 3 through 10) for η = 0.1. Figures 2, 2-12, 2-13, 2-14 and 2-15 are plots of the first four (ℓ=1,2,3,4, respectively) higher order Legendre coefficient curves for η = 0.1. These plots serve to illustrate the degree of significance that higher order terms possess as the scattering acquires a more highly anisotropic character. From Figure 2-15 it can be seen that the magnitudes of the fourth order Legendre coefficients for n=3,4,5,6 are appreciable when compared with their ℓ=0 counterparts of Figure 2-11. A significant difference from the previous η=10.0 case in the shape of the curves is noticable. The n=3 curve peak is higher than the n=4-10 peaks which is not true for η=10.0. Furthermore a shifting of the peaks to the forward direction (source direction) is detectable. These trends become more exaggerated as η decreases.

2.3.4.5 Validation Test of the P_ℓ-Method Computer Code

The reduced source function data produced by the recursive integral equation algorithm (Secs. 2.3.5.2-2.3.5.4) provided a basis for testing the Spencer-Lewis equation integration scheme of the P_ℓ-method code (Secs. 2.2.3, 2.2.4, 2.2.4.2, 2.2.5).

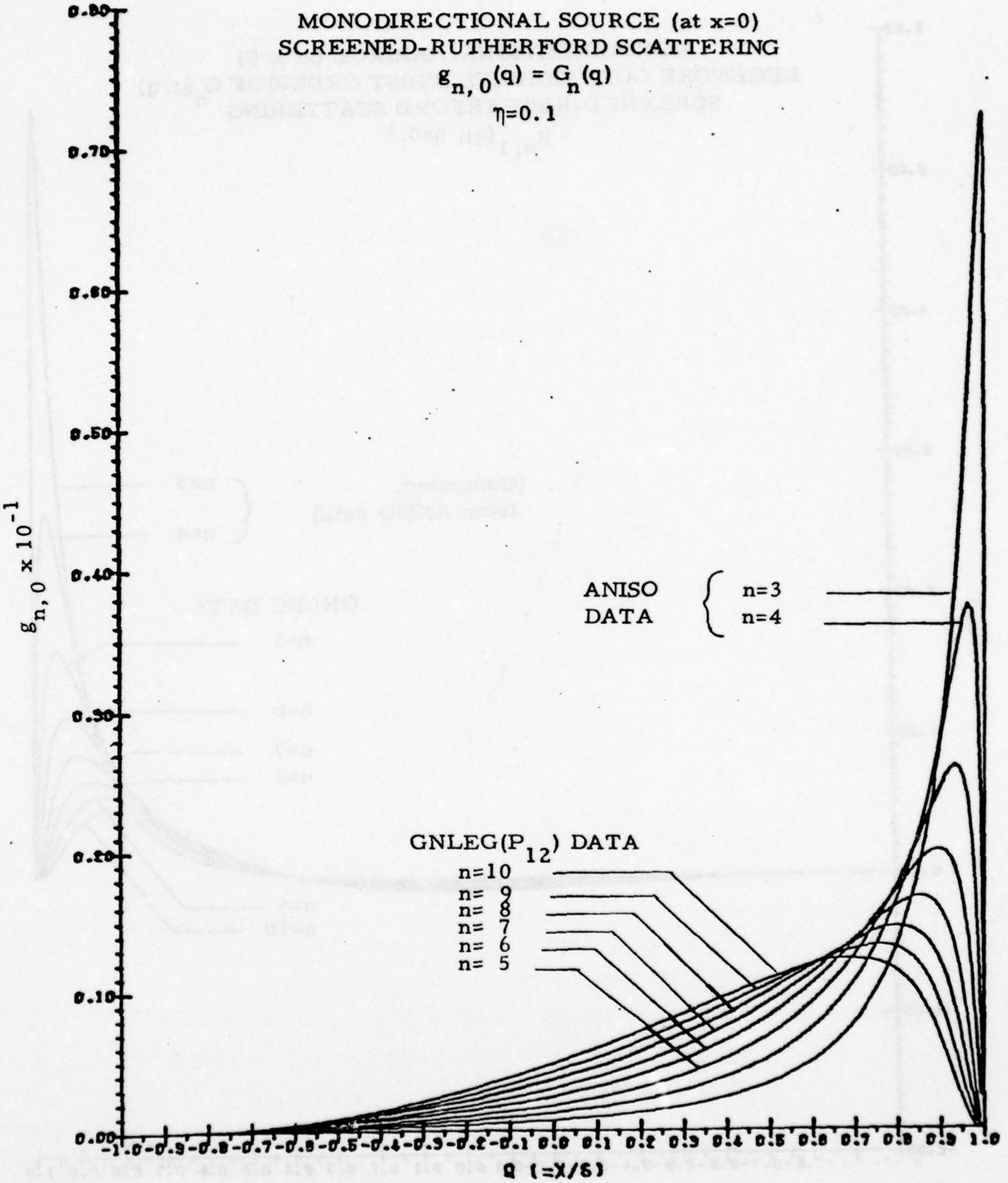

MONODIRECTIONAL SOURCE (at x=0)
SCREENED-RUTHERFORD SCATTERING
$g_{n,0}(q) = G_n(q)$
η=0.1
ANISO DATA
n=3
n=4
GNLEG(P_{12}) DATA
n=10
n= 9
n= 8
n= 7
n= 6
n= 5
$g_{n,0} \times 10^{-1}$
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
-1.0
-0.9
-0.8
-0.7
-0.6
-0.5
-0.4
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Q (=X/S)

Fig. 2-11

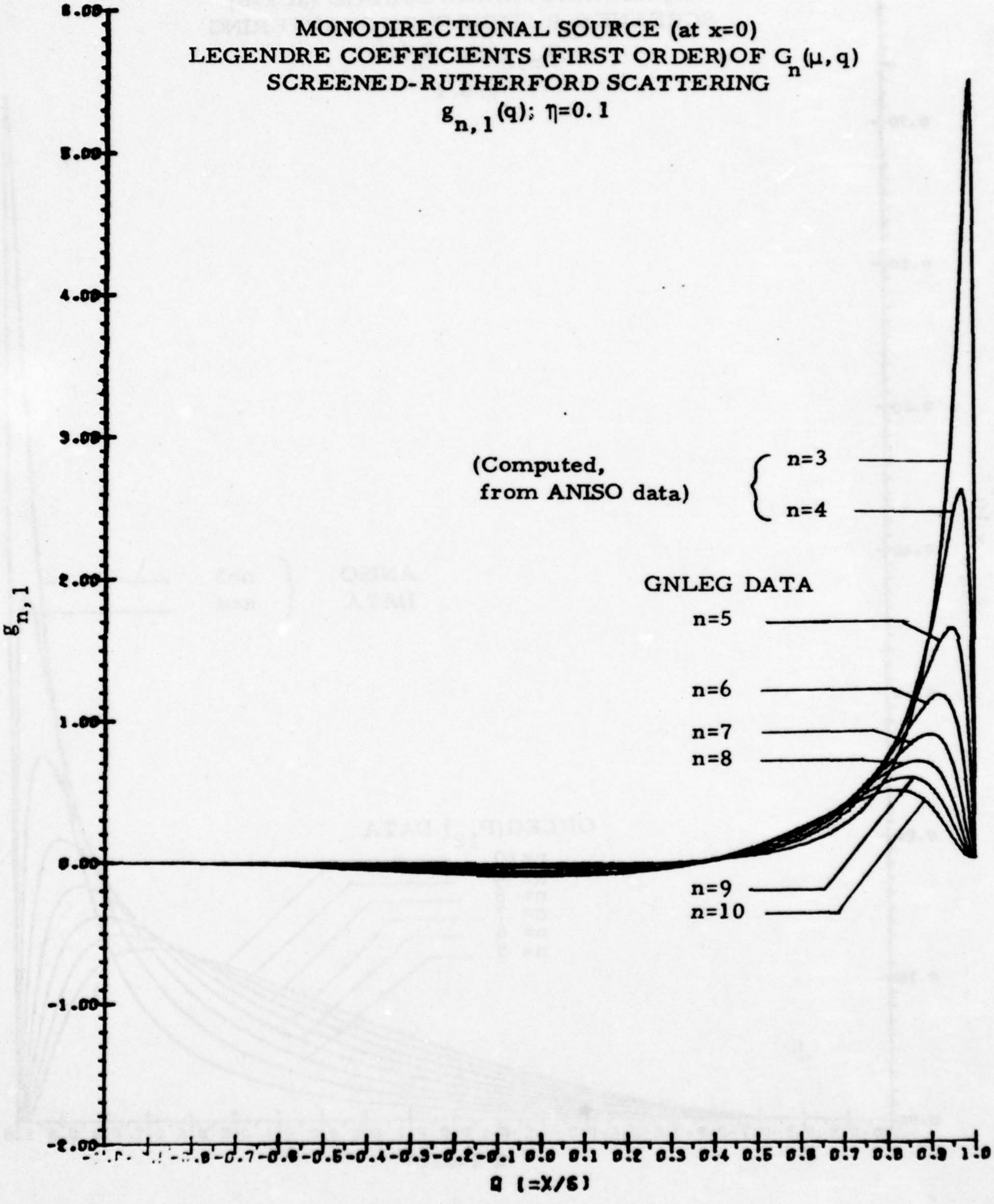

MONODIRECTIONAL SOURCE (at x=0)
LEGENDRE COEFFICIENTS (FIRST ORDER) OF $G_n(\mu, q)$
SCREENED-RUTHERFORD SCATTERING
$g_{n,1}(q)$; η=0.1
(Computed, from ANISO data)
n=3
n=4
GNLEG DATA
n=5
n=6
n=7
n=8
n=9
n=10
6.00
5.00
4.00
3.00
2.00
1.00
0.00
-1.00
-2.00
$g_{n,1}$
-0.8
-0.7
-0.6
-0.5
-0.4
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Q (=X/S)

Fig. 2-12

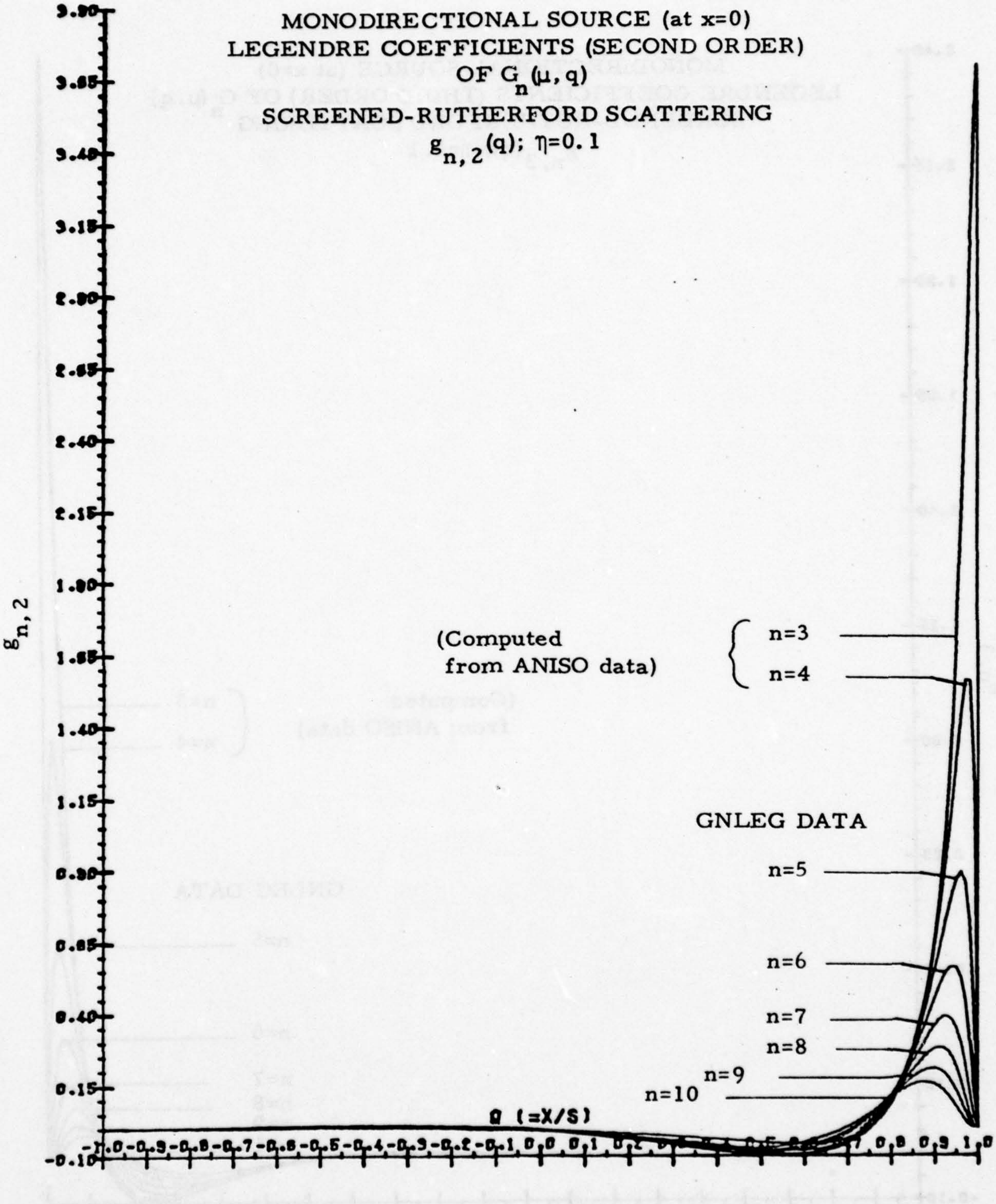

MONODIRECTIONAL SOURCE (at x=0)
LEGENDRE COEFFICIENTS (SECOND ORDER)
OF $G_n(\mu, q)$
SCREENED-RUTHERFORD SCATTERING
$g_{n,2}(q)$; $\eta=0.1$
(Computed from ANISO data)
n=3
n=4
GNLEG DATA
n=5
n=6
n=7
n=8
n=9
n=10
$g_{n,2}$
Q (=X/S)
3.90
3.65
3.40
3.15
2.90
2.65
2.40
2.15
1.90
1.65
1.40
1.15
0.90
0.65
0.40
0.15
-0.10
-1.0 -0.9 -0.8 -0.7 -0.6 -0.5 -0.4 -0.3 -0.2 -0.1 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

Fig. 2-13

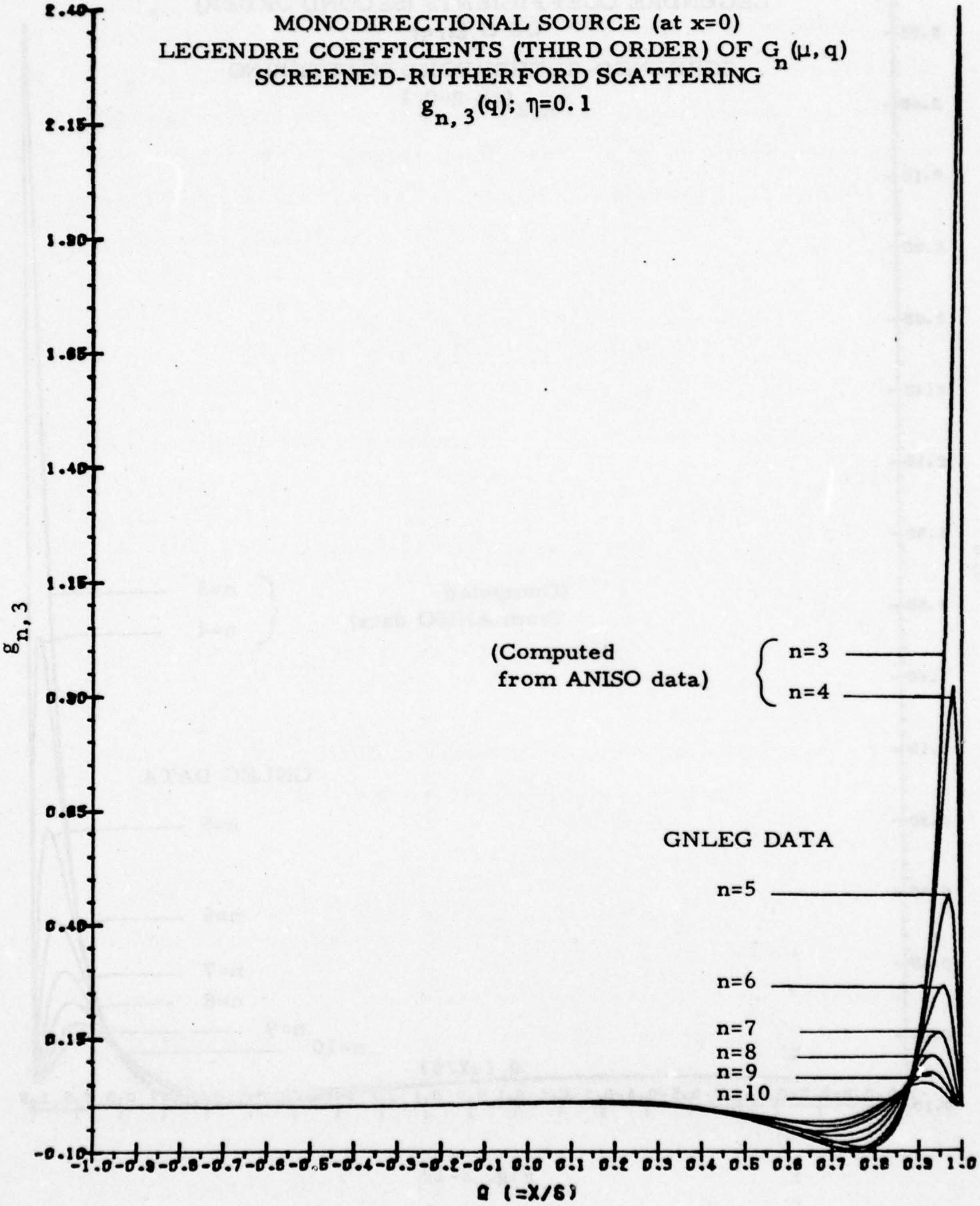

MONODIRECTIONAL SOURCE (at x=0)
LEGENDRE COEFFICIENTS (THIRD ORDER) OF $G_n(\mu, q)$
SCREENED-RUTHERFORD SCATTERING
$g_{n,3}(q)$; η=0.1
$g_{n,3}$
2.40
2.15
1.90
1.65
1.40
1.15
0.90
0.65
0.40
0.15
-0.10
-1.0
-0.9
-0.8
-0.7
-0.6
-0.5
-0.4
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Q (=X/S)
(Computed
from ANISO data)
n=3
n=4
GNLEG DATA
n=5
n=6
n=7
n=8
n=9
n=10

Fig. 2-14

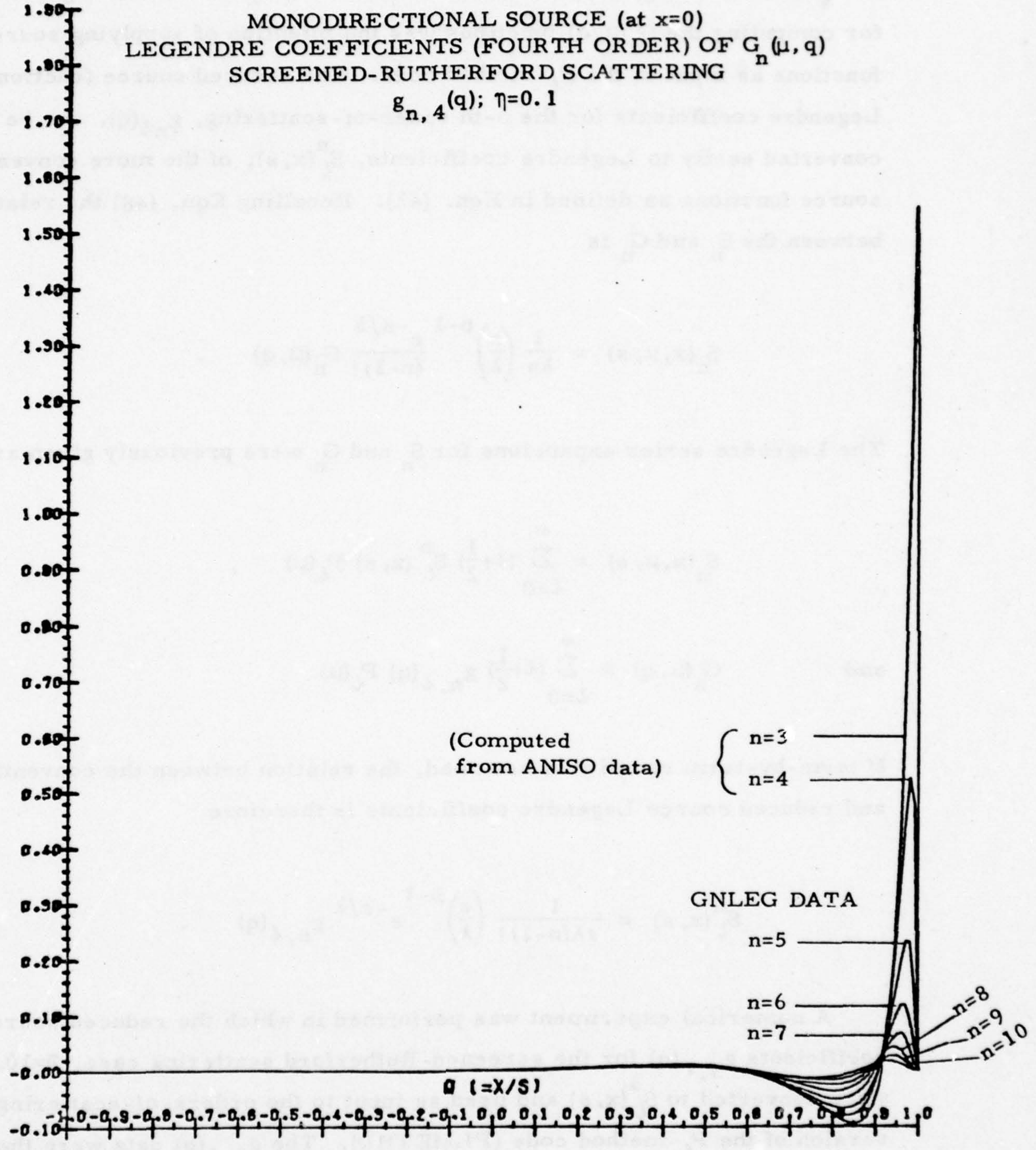

MONODIRECTIONAL SOURCE (at x=0)
LEGENDRE COEFFICIENTS (FOURTH ORDER) OF $G_n(\mu, q)$
SCREENED-RUTHERFORD SCATTERING
$g_{n,4}(q)$; η=0.1
$g_{n,4}$
1.90
1.80
1.70
1.60
1.50
1.40
1.30
1.20
1.10
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
-0.00
-0.10
(Computed from ANISO data)
n=3
n=4
GNLEG DATA
n=5
n=6
n=7
n=8
n=9
n=10
Q (=X/S)
-1.0 -0.9 -0.8 -0.7 -0.6 -0.5 -0.4 -0.3 -0.2 -0.1 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

Fig. 2-15

One of the original motivations for the development of the algorithms for computing the $G_n(\mu, q)$ functions was the intention of supplying source functions as input to the P_ℓ-method code. The reduced source function Legendre coefficients for the n-th order-of-scattering, $g_{n\ell}(q)$, can be converted easily to Legendre coefficients, $S_\ell^n(x, s)$, of the more conventional source functions as defined in Eqn. (42). Recalling Eqn. (48) the relation between the S_n and G_n is

$$S_n(x,\mu,s) = \frac{1}{\lambda s}\left(\frac{s}{\lambda}\right)^{n-1} \frac{e^{-s/\lambda}}{(n-1)!} G_n(\mu,q) \quad . \qquad (48)$$

The Legendre series expansions for S_n and G_n were previously given as

$$S_n(x,\mu,s) = \sum_{\ell=0}^{\infty} (\ell+\tfrac{1}{2})\, S_\ell^n(x,s)\, P_\ell(\mu) \quad , \qquad (9^*)$$

and

$$G_n(\mu,q) = \sum_{\ell=0}^{\infty} (\ell+\tfrac{1}{2})\, g_{n,\ell}(q)\, P_\ell(\mu) \qquad (81)$$

If term-by-term equality is assumed, the relation between the conventional and reduced source Legendre coefficients is therefore

$$S_\ell^n(x,s) = \frac{1}{s\lambda(n-1)!}\left(\frac{s}{\lambda}\right)^{n-1} e^{-s/\lambda}\, g_{n,\ell}(q) \quad . \qquad (88)$$

A numerical experiment was performed in which the reduced source coefficients $g_{3,\ell}(q)$ for the screened-Rutherford scattering case, η=10.0, were converted to $S_\ell^3(x, s)$ and used as input to the orders-of-scattering version of the P_ℓ-method code (PLMETHD). The $g_{3,\ell}(q)$ data were those

[*] Orders-of-scattering version of Eqn. (9).

shown in Figures 2-6 through 2-10 (n=3 curves) and were obtained by resolving the results of the ANISO calculations, $G_3(\mu, q)$, into their Legendre coefficients $g_{3,\ell}(q)$; $\ell=0,1,2,3,4$. The P_ℓ-method code was then used to compute the flux function Legendre coefficients, $f_\ell^3(x, s)$, defined by Eqn. (43), using the integration scheme of Eqn. (39).

The Legendre coefficients of the source function $S_\ell^4(x, s)$ for the next order-of-scattering, n=4, were then computed directly from the $f_\ell^3(x, s)$ using a simple relation derived in the following way:

Starting with Eqn. (46),

$$S_n(x, \mu, s) = \frac{1}{\lambda} \int_{-1}^{1} P(\mu' \to \mu) f_{n-1}(x, \mu', s) \, d\mu' \quad , \tag{46}$$

and expanding the flux

$$f_{n-1}(x, \mu', s) = \sum_{\ell=0}^{\infty} (\ell+\frac{1}{2}) f_\ell^{n-1}(x, s) P_\ell(\mu') \quad , \tag{8*}$$

then if Eqn. (9*), Eqn. (46) and Eqn. (82) are combined, one has

$$\sum_{\ell=0}^{\infty} (\ell+\frac{1}{2}) S_\ell^n(x, s) P_\ell(\mu) =$$

$$\frac{1}{\lambda} \sum_{\ell=0}^{\infty} (\ell+\frac{1}{2}) f_\ell^{n-1}(x, s) \sum_{j=0}^{\infty} (j+\frac{1}{2}) d_j P_j(\mu) \int_{-1}^{1} d\mu' \, P_\ell(\mu') P_j(\mu') \quad .$$

When the orthogonality condition of the Legendre polynomials is invoked, the result is (assuming term-by-term equality)

$$S_{\ell}^{n}(x,s) = \frac{d_{\ell}}{\lambda} f_{\ell}^{n-1}(x,s) \quad . \tag{89}$$

The reduced source function coefficients $g_{4,\ell}(q)$ were then computed from the $S_{\ell}^{4}(x,s)$ at $s = 1.0$ and then plotted. The $S_{\ell}^{4}(x,s)$ function was, in turn, used as input to the P_{ℓ}-method code, and the above described cycle was repeated six times until all of the $g_{n,\ell}(q)$ curves, n=4-10, ℓ=0-4, were computed by the P_{ℓ}-method code. Plots of the curve family $g_{n,0}(q)$, n=3-10, computed for s=1.0 with the mean-free-path taken as λ=0.1, are shown in Figure 2-16. From this plot, it can be seen that the results obtained by the P_{ℓ}-method integration of the Spencer-Lewis equation are extremely close to those obtained by the integral equation method (Fig. 2-6). Similar agreement was obtained for the higher order ℓ terms. The agreement between the two methods leads to a favorable conclusion regarding the validity of the P_{ℓ}-method integration scheme.

2.3.5 Point Isotropic Source; Screened-Rutherford Scattering

The integral recursion equation, Eqn. (59), for the reduced source functions $G_n(\mu, q)$ was derived independently of source geometry considerations. To exploit the flexibility of this method, a series of computer runs of the ANISO and GNLEG codes was performed for a second source geometry, the point isotropic source, with screened-Rutherford scattering. A further motivation for performing these calculations was provided by the fact that the G_2 and G_3 functions for the point isotropic source geometry in an isotropically scattering medium could be calculated by another method, that of Ganapol and Grossman[10]. Thus a comparison of the G_n functions computed via Eqn. (59) with the same quantities calculated using a completely independent method was possible, since isotropic scattering could be closely simulated with the screened-Rutherford scattering formula if η were chosen sufficiently large.

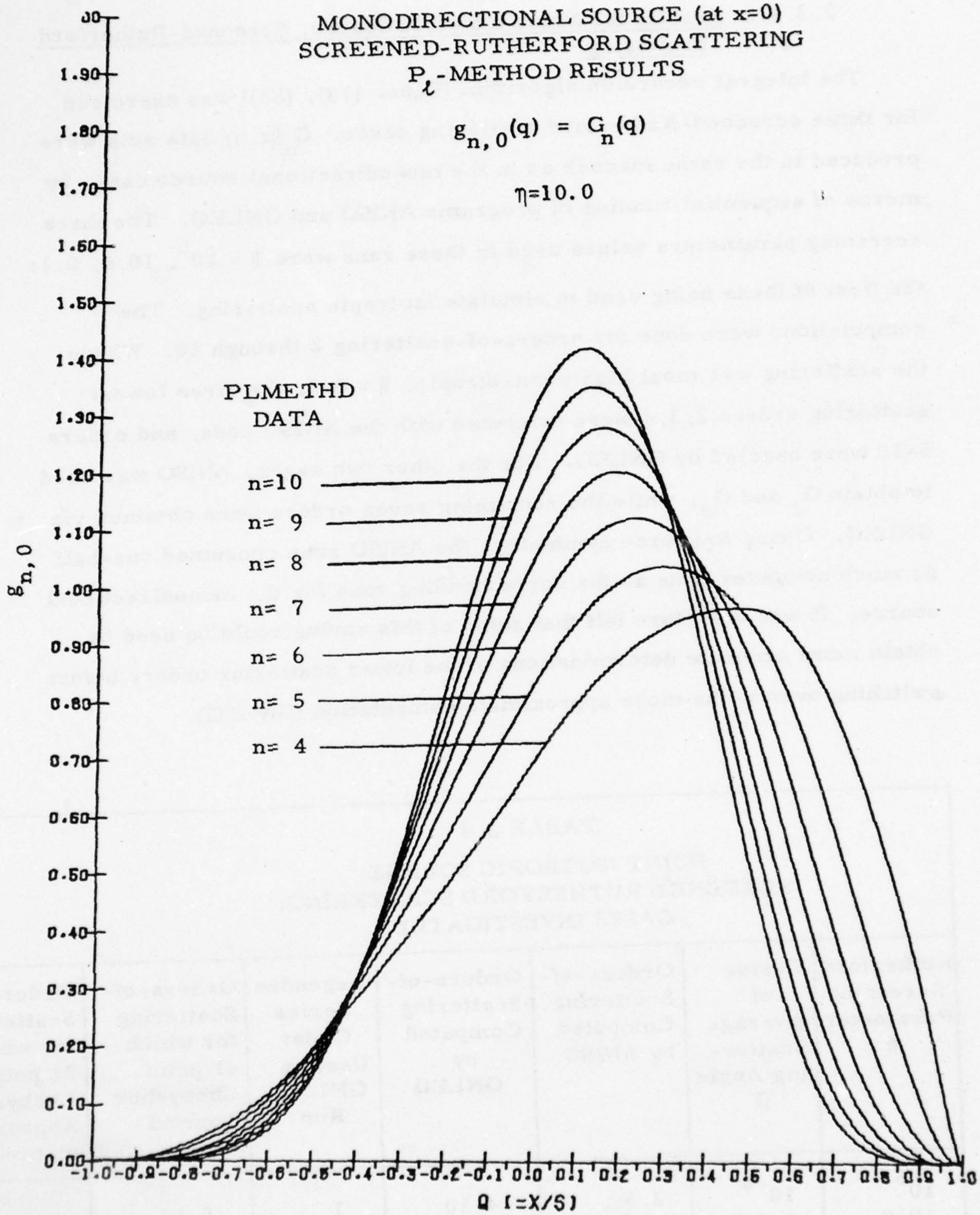

MONODIRECTIONAL SOURCE (at x=0)
SCREENED-RUTHERFORD SCATTERING
P_ℓ-METHOD RESULTS
$g_{n,0}(q) = G_n(q)$
η=10.0
PLMETHD
DATA
n=10
n= 9
n= 8
n= 7
n= 6
n= 5
n= 4
$g_{n,0}$
Q (=X/S)
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
1.10
1.20
1.30
1.40
1.50
1.60
1.70
1.80
1.90
-1.0
-0.9
-0.8
-0.7
-0.6
-0.5
-0.4
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

Fig. 2-16

2.3.5.2 Results for Point-Isotropic Source, Screened-Rutherford Scattering

The integral recursion algorithm (Eqns. (78), (83)) was exercised for three screened-Rutherford scattering cases. $G_n(\mu, q)$ data sets were produced in the same manner as in the monodirectional source case, by means of sequential running of programs ANISO and GNLEG. The three screening parameters values used in these runs were $\eta = 10^6$, 10.0, 0.1; the first of these being used to simulate isotropic scattering. The computations were done for orders-of-scattering 2 through 10. Where the scattering was most highly anisotropic, $\eta = 0.1$, the three lowest scattering orders 2, 3, 4 were computed with the ANISO code, and orders 5-10 were handled by GNLEG. For the other two cases, ANISO was used to obtain G_2 and G_3, while the remaining seven orders were obtained via GNLEG. Owing to source symmetry, the ANISO runs consumed one-half as much computer time as the corresponding runs for the monodirectional source. It was therefore felt that some of this saving could be used to obtain more accurate determinations of the lower scattering orders before switching over to the more approximate computation (GNLEG).

TABLE 2.4

POINT ISOTROPIC SOURCE;
SCREENED RUTHERFORD SCATTERING;
CASES INVESTIGATED

Rutherford Screening Parameter η	Cosine of Average Scattering Angle $\bar{\mu}$	Orders-of-Scattering Computed by ANISO	Orders-of-Scattering Computed by GNLEG	Legendre Series Order Used in GNLEG Run	Orders-of-Scattering for which 41 point Chebyshev Approximation used	Orders-of-Scattering for which 21 point Chebyshev Approximation used
10^6	10^{-6}	2, 3	4-10	1	5-10	-
10.0	0.091	2, 3	4-10	3	5-10	-
0.1	0.909	2, 3, 4	5-10	12	6, 7	8, 9, 10

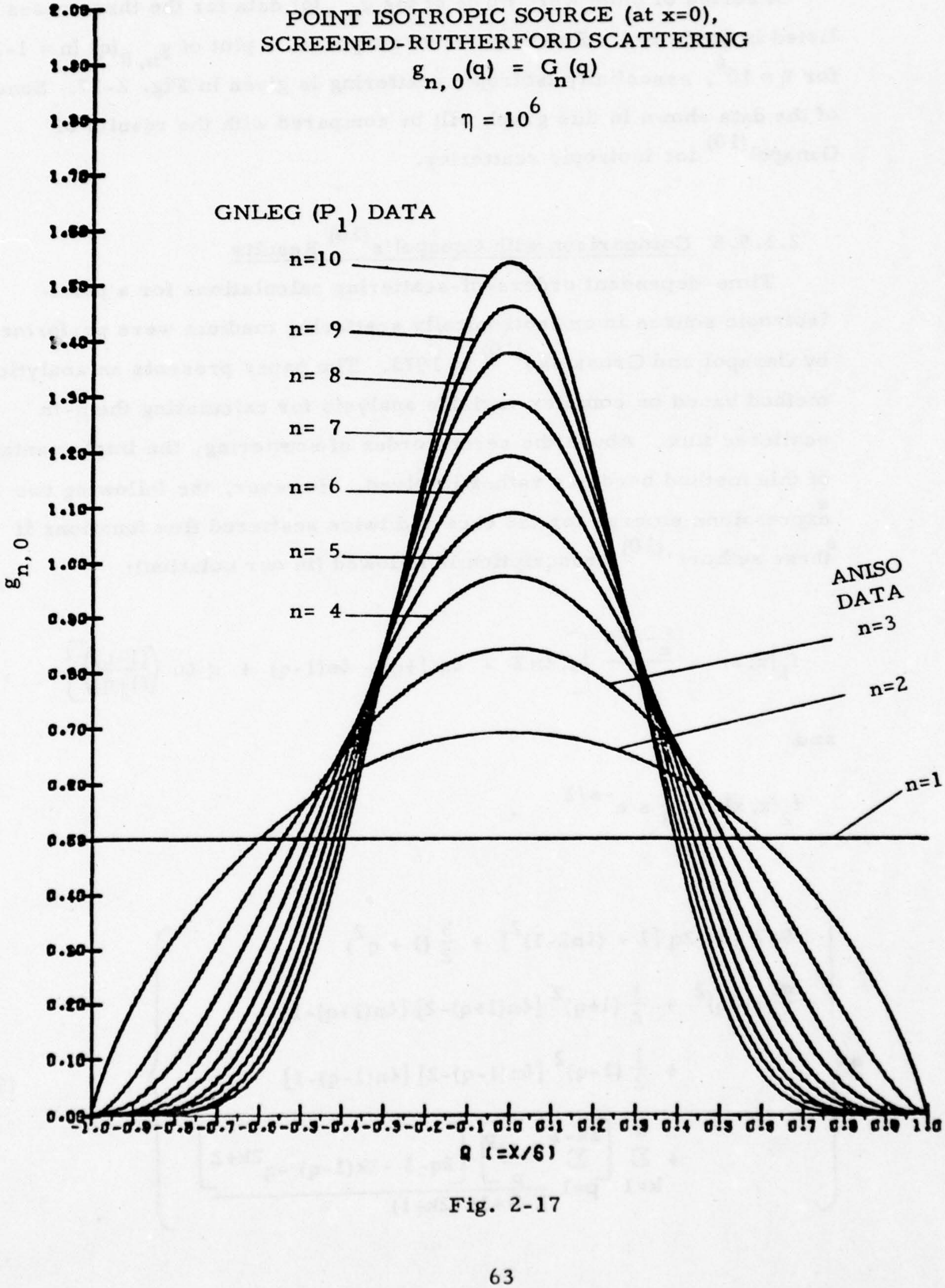

POINT ISOTROPIC SOURCE (at x=0),
SCREENED-RUTHERFORD SCATTERING
$g_{n,0}(q) = G_n(q)$
$\eta = 10^6$
GNLEG (P_1) DATA
n=10
n= 9
n= 8
n= 7
n= 6
n= 5
n= 4
ANISO DATA
n=3
n=2
n=1
$g_{n,0}$
2.00
1.90
1.80
1.70
1.60
1.50
1.40
1.30
1.20
1.10
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
-1.0 -0.9 -0.8 -0.7 -0.6 -0.5 -0.4 -0.3 -0.2 -0.1 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
Q (=X/S)

Fig. 2-17

A series of plots were made of the $g_{n,\ell}(q)$ data for the three cases listed in Table 2.4. For illustrative purposes, a plot of $g_{n,0}(q)$ (n = 1-10) for $\eta = 10^6$, essentially isotropic scattering is given in Fig. 2-17. Some of the data shown in this graph will be compared with the results of Ganapol(10) for isotropic scattering.

2.3.5.3 Comparison with Ganapol's(10) Results

Time-dependent orders-of-scattering calculations for a point-isotropic source in an isotropically scattering medium were performed by Ganapol and Grossman(10) in 1973. The paper presents an analytical method based on complex variable analysis for calculating the n-th scattered flux. Above the second order of scattering, the implementation of this method becomes rather involved. However, the following two expressions emerge for the once and twice scattered flux functions if these authors'(10) prescription is followed (in our notation):

$$f_1(x, s) = \frac{e^{-s/\lambda}}{2} \left[2 \ln 2 - \ln(1+q) - \ln(1-q) + q \ln \left(\frac{(1-q)}{(1+q)}\right) \right] \quad , \tag{90}$$

and

$$f_2(x, s) = \frac{3}{8} s\, e^{-s/\lambda} \quad .$$

$$\bullet \left\{ \begin{aligned} & 2 \ln 2 - 3 - 2q \left[1 + (\ln 2 - 1)^2\right] + \frac{3}{2}(1 + q^2) \\ & - \frac{\pi 2}{6}(1-q)^2 + \frac{1}{2}(1+q)^2 \left[\ln(1+q) - 2\right]\left[\ln(1+q) - 1\right] \\ & \qquad + \frac{1}{2}(1-q)^2 \left[\ln(1-q) - 2\right]\left[\ln(1-q) - 1\right] \\ & \qquad + \sum_{k=1}^{\infty} \frac{\left[\sum\limits_{p=1}^{2k-1} \frac{(-1)^p}{p}\right] \left[2q - 1 - 2k(1-q) - q^{2k+2}\right]}{k\,(k+1)\,(2k+1)} \end{aligned} \right\} \quad . \tag{91}$$

The above quantities, f_1 and f_2, can be transformed into reduced fluxes readily enough via Eqn. (47). Furthermore, the reduced source results, $G_n(q)$, obtained by application of the integral recursion formula, Eqn. (59), for $\eta=10^6$ can also be transformed into reduced flux functions. If the relation between the reduced source and reduced flux functions is integrated over all possible final cosine values, μ, the following results;

$$\begin{aligned} G_n(q) &= \int_{-1}^{1} d\mu \, G_n(\mu, q) \\ &= \int_{-1}^{1} d\mu \int_{-1}^{1} d\mu' \, P(\mu' \rightarrow \mu) \, F_{n-1}(\mu', q) \\ &= \int_{-1}^{1} d\mu' \, F_{n-1}(\mu', q) \\ &= F_{n-1}(q) \, , \end{aligned} \tag{92}$$

since

$$\int_{-1}^{1} d\mu \, P(\mu' \rightarrow \mu) = 1 \quad . \tag{93}$$

From the above, it is clear that the values obtained here for $G_2(q)$ and $G_3(q)$, with $\eta=10^6$, should closely match $F_1(q)$ and $F_2(q)$, respectively, obtained with the algorithm of Ref. (10). Table 2.5 presents a comparison of these quantities.

TABLE 2.5

COMPARISON OF REDUCED SOURCE FUNCTIONS WITH RESULTS OF GANAPOL AND GROSSMAN[10]

q*	F_1 (Computed from Ref. 10)	G_2 ($\eta=10^6$)	F_2 (Computed from Ref. 10)	G_3 ($\eta=10^6$)
0.0	.69316	.69277	.84520	.84408
0.1	.68814	.68776	.83293	.83178
0.2	.67301	.67262	.79640	.79519
0.3	.64745	.64704	.73645	.73524
0.4	.61087	.61044	.65472	.65358
0.5	.56233	.56188	.55385	.55274
0.6	.50040	.49992	.43748	.43643
0.7	.42272	.42220	.31108	.31012
0.8	.32508	.32455	.18301	.18219
0.9	.19852	.19795	.06761	.06705
1.0	0.	0.	0.	0.

*Only positive q values are given here since the F and G functions are symmetric in q.

2.4 REFERENCES

1. Garth, J. C., O'Brien, J. V., "Numerical Solution of the Spencer-Lewis Equation for Kilovolt Electron Transport in Plane Slabs," IEEE Trans. Nuc. Sci., NS-20, 82, (1974).

2. Berger, M. J., and Seltzer, S. M., National Academy of Sciences—National Research Council, Publication 1133, 69 (1964).

3. Spencer, L. V., N.B.S. Monograph 1, "Energy Dissipation by Fast Electrons".

4. Clark, M. and Hansen, K. F., Numerical Methods of Reactor Analysis, Academic Press, 184 (1964).

5. Weinberg, A. M. & Wigner, E. P., The Physical Theory of Neutron Chain Reactors, U. of Chicago Press, 257 (1958).

6. Garth, J. C., Private Communication.

7. Richtmeyer, R. D. and Morton, K. W., Difference Methods for Initial Value Problems, John Wiley & Sons, 233, (1967).

8. Mitchell, A. R., Computational Methods in Partial Differential Equations, Wiley-Interscience, 159 (1969).

9. Street, R. L., Analysis and Solution of Partial Differential Equations, Brooks, Cole Publishing Co., Monterey, California, 304 (1973).

10. Ganapol, B. D. and Grossman, L. M., "The Collided Flux Expansion Method for Time Dependent Neutron Transport," Nuc. Sci. & Eng., 52, 454 (1953).

11. Ganapol, B. D., McKenty, P. W. and Peddicord, K. L., "The Generation of Time-Dependent Neutron Transport Solutions in Infinite Media," Nucl. Sci. Eng. 64, 317 (1977).

12. Gradshteyn, I. S. and Rhyzhik, I. M., Tables of Integrals, Series, and Products, Academic Press, 66 (1965).

13. Lewin, L., Dilogarithms and Associated Functions, MacDonald, pp. 1-23 (1956).

14. Ginsberg, E. S., Communications ACM, 18, 4, P200 (1975).

15. Prenter, P. M., Splines and Variational Methods, Wiley Interscience, pp. 53-57, 130 (1975).

3. SAW Device Investigations

3.1 Introduction

During the period covered by this contract, our efforts were directed towards utilizing and extending the analyses and programs developed previously for the investigation and design of SAW transducers which began under USAF Contract No. F19628-74-C-0049. A comprehensive tabulation of the salient data and plots resulting from this body of work can be found elsewhere;** however, a representative example of the work performed involving a 3-phase IDT and a listing of the most important studies executed is presented here. Since the key programs required for these efforts are also expected to form the nucleus of tools for future work, special attention has been given to the development of user-oriented documentation for their operation.

3.2 3-Phase Transducers

The 3-phase IDT is of considerable practical interest because it has the desirable property of radiating most of its SAW energy in one direction along a substrate surface. With a view towards designing this class of transducer it was necessary that the acoustic responses of the triplets of electrodes comprising such devices be obtained.

To obtain quantitative benchmarks for such work, a test case was analyzed in which a full array of 13 triplets (39 electrodes) was subjected to complex potentials. Our previous efforts had only required formulations in terms of real potentials, but numerical results calculated by others for the response of 3-phase transducers have thus far been based on complex potentials. The analysis and programming operations required for this investigation serve to illustrate the manner in which this and the other studies were approached and undertaken.

**To appear in a forthcoming in-house report.

3.3 Theory

The electric displacement, D(x'), normal to the interface between an array of electrodes and a piezoelectric substrate is related to the (real) potentials, V(x), applied to the electrodes through a linear relationship:

$$V(x) = \mathbf{L}\{D(x')\} \quad , \tag{1}$$

where $\mathbf{L}$ is the appropriate linear, Fredholm integral operator of the first kind[(1)] governing electrostatic interactions among line charges residing on infinitely thin electrodes. Symbolically,

$$D(x') = \mathbf{L}^{-1}\{V(x)\} \quad . \tag{2}$$

The Fourier transform operator, $\mathbf{F}$, yields

$$\begin{aligned} \hat{D}(k) &= \mathbf{F}\{D(x')\} \\ &= \mathbf{F}\mathbf{L}^{-1}\{V(x)\} \\ &\equiv \hat{D}_R(k) + i\,\hat{D}_I(k) \end{aligned} \tag{3}$$

In view of the linearity of the operators it follows that for the complex potential

$$V(x) \equiv V_a(x) + i\,V_b(x) \quad , \tag{4}$$

$$\begin{aligned} \hat{D}(k) &= \mathbf{F}\mathbf{L}^{-1}\{V(x)\} \\ &= \mathbf{F}\mathbf{L}^{-1}\{V_a(x)\} + i\,\mathbf{F}\mathbf{L}^{-1}\{V_b(x)\} \\ &= \hat{D}^a(k) + i\,\hat{D}^b(k) \\ &= \hat{D}^a_R(k) + i\,\hat{D}^a_I(k) + i\left[\hat{D}^b_R(k) + i\,\hat{D}^b_I(k)\right] \\ &= \hat{D}^a_R(k) - \hat{D}^b_I(k) + i\left[\hat{D}^a_I(k) + \hat{D}^b_R(k)\right] \quad . \end{aligned} \tag{5}$$

With the acoustic response of an electrode or group of electrodes given by

$$A(k) \equiv \sqrt{|k|}\ D(k) \quad , \tag{6}$$

it is clear that A(k) can be found for a complex potential by determining the D-fields due to $V_a^{(x)}$ and $V_b^{(x)}$ separately and linearly combining the results as indicated above. The problem thus became one of obtaining expressions for $D^a(k)$ and $D^b(k)$ in terms of $V_a^{(x)}$ and $V_b^{(x)}$. This was accomplished in several stages:

1. $\{V_a\}$ was inputted to program GAL3AT which then computed surface charge density expansion coefficients, ${}_aC_n^{(j)}$.
2. $\{V_b\}$ was also inputted to GAL3AT and yielded ${}_bC_n^{(j)}$.
3. The expansion coefficients were analytically related to $\hat{D}^a(k)$ and
4. Program TRPL was then used to evaluate $\hat{D}^a(k)$, $\hat{D}^b(k)$, and A(k).

The complex potentials applied to the 3-phase array were defined as follows:

$$V_j \equiv e^{i(j-1)120^\circ} \qquad j = 1\,(1)\,39 \tag{7}$$

with
$$V_a^{(j)} = \cos\,(j-1)\ 120^\circ \tag{8}$$

and
$$V_b^{(j)} = \sin\,(j-1)\ 120^\circ\ . \tag{9}$$

GAL3AT yielded $\{{}_aC_n^{(j)}\}$ and $\{{}_bC_n^{(j)}\}$. The relationship between these coefficients and $\hat{D}(k)$ will now be derived.

3.4 Derivation of $\hat{D}(k)$ and $\hat{D}(\nu, \eta)$

For the middle triplet of electrodes, the Fourier transform of the normal component of the electric displacement under the electrodes is given by

$$D(k) \equiv \int_{L-s/2}^{L+s/2} D(x')\, e^{ikx'}\, dx' \tag{10}$$

$$= \int_{-L-s/2}^{-L+s/2} D_-(x')\, e^{ikx'}\, dx' + \int_{L-s/2}^{L+s/2} D_+(x')\, e^{ikx'}\, dx'$$

$$+ \int_{-s/2}^{s/2} D_0(x')\, e^{ikx'}\, dx' .$$

where $D_-(x')$, $D_0(x')$, and $D_+(x')$ are the electric displacements at the surfaces of the left, middle, and right electrodes of the central triplet. Between the electrodes there is no free charge; hence, $D(x')$ in these regions is zero.

If the D-field under each electrode is expanded in terms of the normalized spatial variable u, namely as

$$D(u) \equiv (1 - u^2)^{-1/2} \sum_{n=0}^{N-1} C_n T_n(u)$$

and substituted into (10), D(k) becomes equal to

$$D(k) = \frac{s}{2} e^{ikL} \sum_{n=0}^{N-1} C_n^{(+)} \int_{-1}^{1} (1 - u_1^2)^{-1/2} \quad T_n(u_1)\, e^{ik\frac{s}{2}u_1}\, du_1$$

$$+ \frac{s}{2} e^{-ikL} \sum_{n=0}^{N-1} C_n^{(-)} \int_{-1}^{1} (1 - u_2^2)^{-1/2} \quad T_n(u_2)\, e^{ik\frac{s}{2}u_2}\, du_2$$

$$+ \frac{s}{2} \sum_{n=0}^{N-1} C_n^{(0)} \int_{-1}^{1} (1 - u_3^2)^{-1/2} \quad T_n(u_3)\, e^{ik\frac{s}{2}u_3}\, du_3$$

where $u_1 = (x' - L)/(s/2)$

$u_2 = (x' + L)/(s/2)$

$u_3 = x'/(s/2)$

$T_n(u) =$ n^{th} order Chebyshev polynomial of the first kind

$s =$ width of an electrode

$L =$ center-center separation between adjacent electrodes

$C_n^{(+,-,0)} =$ (unknown) expansion coefficients of the electric displacement under the +, -, or 0 electrode.

With the change of variables $u \equiv \cos\theta$, integrals of the form

$$\int_0^{\pi} \cos\theta \; e^{ik\frac{s}{2}\cos\theta} \; d\theta = \pi i^n \; J_n(k\,s/2) \tag{13}$$

result, where $J_n(\cdot)$ is the n^{th} order Bessel function of the first kind. Collecting terms yields

$$\begin{aligned} \frac{2}{\pi s}\hat{D}(k) = {} & e^{ikL} \sum_{n=0}^{N-1} C_n^{(+)} i^n \; J_n(k\,s/2) \\ & + e^{-ikL} \sum_{n=0}^{N-1} C_n^{(-)} i^n \; J_n(k\,s/2) \\ & + \sum_{n=0}^{N-1} C_n^{(0)} i^n \; J_n(k\,s/2) \end{aligned} \tag{14}$$

At this point it is useful to relate the fundamental wavelength, λ_0, of the acoustic waves to the electrode spacing, L. For a 3-phase device $\lambda_0 = 3L$, and it follows that

$$\begin{aligned} k\,s/2 &= k\,L\,s/(2L) = k\,L\,\eta/2 \\ &\equiv k_0\,\nu\,L\,\eta/2 \\ &= (2\pi/\lambda_0)\,\nu\,L\,\eta/2 \\ &= \pi\,\eta\,\nu/3 \end{aligned} \tag{15}$$

where ν is a convenient dimensionless, normalized parameter: $\nu = f/f_0 = \lambda_0/\lambda$. Inserting this into the previous formula for $\hat{D}(k)$ yields an expression in terms of the more convenient parameters η and ν:

$$\hat{D}(\nu,\eta) = \hat{D}_R(\nu,\eta) + i\,\hat{D}_I(\nu,\eta) \tag{16}$$

where

$$\hat{D}_R(\nu,\eta) = \sum_{n=0}^{(N-1)/2} (-1)^n \Bigl\{ J_{2n}\Bigl(\frac{\pi}{3}\eta\nu\Bigr) \Bigl[C_{2n}^{(0)} + \Bigl(C_{2n}^{(+)} + C_{2n}^{(-)} \Bigr) \cos\frac{2\pi}{3}\nu \Bigr] \tag{17}$$

$$+ J_{2n+1}\Bigl(\frac{\pi}{3}\eta\nu\Bigr) \Bigl[C_{2n+1}^{(-)} - C_{2n+1}^{(+)} \Bigr] \sin\frac{2\pi}{3}\nu \Bigr\}$$

and

$$\hat{D}_I(\nu,\eta) = \sum_{n=0}^{(N-1)/2} (-1)^n \Bigl\{ J_{2n}\Bigl(\frac{\pi}{3}\eta\nu\Bigr) \Bigl[C_{2n}^{(+)} - C_{2n}^{(-)} \Bigr] \sin\frac{2\pi}{3}\nu \tag{18}$$

$$+ J_{2n+1}\Bigl(\frac{\pi}{3}\eta\nu\Bigr) \Bigl(C_{2n+1}^{(+)} + C_{2n+1}^{(-)} \Bigr) \cos\frac{2\pi}{3}\nu$$

$$+ C_{2n+1}^{(0)} \Bigr]\Bigr]\Bigr\}$$

The expressions thus derived become equal to $\hat{D}^a_R(\nu, \eta)$ and $\hat{D}^a_I(\nu, \eta)$ when $\{V^{(j)}_a\}$ is used, and $\hat{D}^b_R(\nu, \eta)$ and $\hat{D}^b_I(\nu, \eta)$ when $\{V^{(j)}_b\}$ is applied to the GAL3AT program.

3.4.1 Programming and Associated Tasks

Prior to running program GAL3AT for the determination of the $\{C^{(j)}_n\}$, two types of input must be generated:

a. the sequence of potentials applied to the electrodes, and,

b. the off-diagonal matrix elements relating potentials, $\{V^{(j)}: j=1, 2, \ldots, J\}$, to charge density expansion coefficients, $\{C^{(j)}_n: n=0, 1, \ldots, NN-1; j=1, 2, \ldots, J\}$.

For runs involving a small number of different polarity sequences, the $\{V^{(j)}\}$ were usually punched onto NAMELIST cards that were then read by GAL3AT, whereas for production runs requiring many sequences it was more convenient to create and store each sequence on a disk file that GAL3AT could then access and read.

In the case of the 3-phase transducer, two sets of potentials were punched onto NAMELIST input cards: the set $\{V^{(j)}_a\} \equiv \{\cos(j-1)\,120^\circ: j=1, 2, \ldots, 39\}$ which was punched as

```
$VSEQ POL(1)=1.,-.5,-.5,1.,-.5,-.5,1.,-.5,-.5,1.,-.5,-.5,1.,-.5,-.5,1.,-.5,-.5,
POL(19)=1.,-.5,-.5,1.,-.5,-.5,1.,-.5,-.5,1.,-.5,-.5,1.,-.5,-.5,1.,-.5,-.5,
POL(37)=1.,-.5,-.5$
```

and the set $\{V^{(j)}_b\} \equiv \{\sin(j-1)\,120^\circ: j=1, 2, \ldots, 39\}$ which was punched on another set of VSEQ NAMELIST cards.

The off-diagonal matrix elements for the 3-phase, 39 electrode array was created by program PSIGEN, the matrix elements corresponding to those previously defined.(1, 2)

3.4.2 Task Sequence for 3-Phase IDT Study

1. Define one or more sequence of potentials, $\{V^{(j)}: j=1,2,\ldots,J\}$, to be applied to the electrodes of an array. Punch the data on NAMELIST input cards or store on disk.

2. Run program PSIGEN to obtain the off-diagonal matrix elements for an array having a specified maximum number of electrodes and η-value.

3. Run a GALERKN type program (GAL3AT in the case of the 3-phase transducer) for the calculation of the $\{C_n^{(j)}\}$. Store the $\{C_n^{(j)}\}$ on punched cards and/or on disk.

4. Use the $\{C_n^{(j)}\}$ as input to an acoustic response program (TRPL in the case of the 3-phase IDT). Print or plot acoustic response information; print normalized values of $|A(\nu_0)|$ for design tables if desired.

3.5 Results

The acoustic response of the central triplet in the array was evaluated for $\eta=0.5$ at $\nu=1$ and $\nu=2$. The ratio $|A(\nu=2)|^2/ |A(\nu=1)|^2$ was computed and found to equal -7.96 db. The only available experimental data we were able to obtain came from work done by Texas Instruments, Inc. where a value of -8db was reported. While this was considered to be excellent verification of our analysis and programs, other values of $|A(\nu)|^2/ |A(\nu=1)|^2$ continue to be sought.

3.6 Tabulation of Major Computational Tasks

The major computational tasks performed during the contract are listed below following the definitions of parameters and variables.

Definitions

$$\sigma_j(u) = (1-u^2)^{-1/2} \sum_{n=0}^{NN-1} C_n^{(j)} T_n(u) \qquad (19)$$

= surface charge density, or normal component of the electric displacement, on the j-th electrode in an IDT array.

u = normalized position on the j-th electrode:

$-1 \leq u \leq 1$

$T_n(u)$ = n-th order Chebyshev polynomial of the first kind.

$C_n^{(j)}$ = charge density expansion coefficients

j = 1(1)J, where J = total number of electrode positions in the array.

NN = total number of terms in the expansion for a given electrode.

η = electrode width/center-to-center separation =0.5 unless otherwise noted.

3.6.1 Single Electrode Arrays

A. Off-diagonal matrix elements for $\{C_n^{(j)}\}$ computations

1. J=17; NN=6
2. J=17; NN=10
3. J=17; NN=6; η = .1 (.1) .9
4. J=21; NN=6

B. $\{C_n^{(j)}\}$ data for

1. Source weighting $J=17;\ j=9;\ NN=6$
2. Full array, alternating polarities $J=17;\ j=1(1)9;\ NN=6$
3. Full array, alternating polarities $J=17;\ j=1(1)9;\ NN=6;\ \eta=.1(.1).9$
4. F. Sandy: 621 cases $J=17;\ NN=6$

    a. 378 middle electrode cases

    b. 243 end electrode cases

5. Slowly converging coefficients $J=15;\ NN=10;\ \eta=.88$
6. SAWDESIGN program: 729 cases $J=21;\ j=9;\ NN=6$

C. Hartmann Tables and/or Acoustic Response

1. $J=17;\ j=9;\ \nu \equiv f/f_o=1$
2. $J=17;\ j=9;\ \nu=5$
3. $J=17;\ j=9;\ \nu=9$
4. $J=21;\ j=1(1)21$ acoustic response

D. Smith Tables - 2 nearest neighbors - J=17

1. Expansion coefficients for $(\eta-.5)^m$ series

    $m=0(1)5;\ \eta=.1(.1).9;\ NN=6$

2. Same as #1 with NN=10 and scaling factor = -.506767 to agree with Smith's work.[3]
3. 13 extra cases
4. 91 extra cases

3.6.2 Double Electrode Arrays

A. $\{C_n^{(j)}\}$ and related data

1. Off-diagonal matrix elements for J=22; NN=6
2. $\{C_n^{(j)}\}$: J=16; NN=10; j=1(1)16
3. " J=22; NN=10; j=1(1)18 (benchmark test)
4. " J=22; 3 nearest pairs variations
5. " J=22; NN=8; for Hartmann table

B. Hartmann Tables and Spectral Weighting Data

1. J=22 (11 pairs); j=11, 12; NN=8; ν=1
2. J=22 (11 pairs); j=11, 12; NN=8; ν=3
3. J=18
4. $\{C_n^{(j)}\}$ and $|A(\nu)/A_{9,10}^{(1)}|$ vs. ν for J=18; j=1(1)10
5. $|A_j(\nu)/A_j(1)|$ vs. ν for J=16

3.6.3 Triple Electrode Arrays

A. $\{C_n^{(j)}\}$ data for 39 element array (J=39)

1. j=19, 20, 21 and j=1(1)9 68 cases
   a. Polarity sequence composed of {1, 0, 0} triplets
   b. Polarity sequence composed of {0, 1, 0} triplets
   c. Polarity sequence composed of {0, 0, 1} triplets
2. 54 extra cases

B. $\{C_n^{(j)}\}$ for special series of triplets using the polarity sequence {-1, 2, -1} with 3 triplets withdrawn from the array.

C. $\{C_n^{(j)}\}$ for 3-phase transducer; J=39; j=19, 20, 21;NN=6.

1. For real part of complex potential

2. For imaginary part of complex potential

D. A(ν) Calculations

1. For $\{C_n^{(j)}\}$ corresponding to real part of complex potentials.

2. $|A(\nu)|$ for 4 cases.

3. A(ν=2)/A(ν=1) for different electrodes in the 39 element array of the first case in D. 2.

3.7 REFERENCES

1. K. Laker, E. Cohen, A. Slobodnik, Jr., "Electric Field Interactions Within Finite Arrays and the Design of Withdrawal Weighted SAW Filters at Fundamental and Higher Harmonics," 1976 Ultrasonics Symposium Proceedings, IEEE CAT. NO. CH1120-5SU, pp. 317-321.

2. RADC-TR-86, Final Technical Report, "Interactive Math Modeling," Feb. 1977, pp. 142-146.

3. W. R. Smith, Jr. and W. F. Pedler, "Fundamental and Harmonic-Frequency Circuit Model Analysis of Interdigital Transducers with Arbitrary Metallization Ratios and Polarity Sequences," IEEE Trans. Microwave Theory Tech., Vol. 23, pp. 853-864, Nov. 1975.

4. Phase-Constrained Guided Modes In Dielectric Slabs

4.1 Statement of the Problem

The purpose of this effort was to determine the refractive index profile, n(x), of a sandwich of uniform, homogeneous, lossless dielectric slabs that would bring the propagation modes of monochromatic electromagnetic waves launched into the guiding structure into phase after traveling a specified distance, b, along the propagation axis. In addition to the phase constraints, several other requirements were imposed on the problem:

1. n(x) should be symmetric about the middle slab and decrease monotonically away from it. See Fig. 4-1.
2. The slabs were to be immersed in air, with the refractive indices of the outer slabs fixed at values of 1.50 while the middle one was to assume the value of 1.53.
3. For the ℓ^{th} propagation mode, characterized by the propagation vector $k_z^{(\ell)} \equiv \frac{2\pi}{\lambda_o} \mu_\ell$, the following "1 μ_ℓ/layer" condition was imposed:

   $$n_\ell < \mu_\ell < n_{\ell+1} \qquad \text{for} \qquad \ell=1,2,\ldots,LL$$

   where LL = the total number of modes supported by the guiding structure.

4.2 Phase Constraints

The phase constraints were formulated by requiring that the phase difference between adjacent modes be an integer multiple of 2π after the modes propagate a distance b. This was expressed by

$$\left(k_z^{(\ell+1)} - k_z^{(\ell)}\right) b = 2\pi I_\ell \qquad (1)$$

or

$$\left(\mu_{\ell+1} - \mu_\ell\right)\left(\frac{b}{\lambda_o}\right) = I_\ell \qquad \ell=1,2,\ldots, LL-1$$

where the I_ℓ are integers.

In view of the integer constraints no obvious method for solving this problem presented itself. The technique finally developed was a simple variant of the Newton-Raphson algorithm and will now be described.

A refractive index profile $\{n_i^o, \xi_i^o\}$ was initially assumed, and the eigenvalues $\{\mu_\ell^o\}$ were computed for it. Substituting the μ_ℓ^o into (1) leads to the following

$$\frac{b}{\lambda_o}\left[\mu_{\ell+1}^o - \mu_\ell^o\right] = I_\ell + \epsilon_\ell \qquad \ell=1,2,\ldots, LL-1 \tag{2}$$

where $\epsilon_\ell \neq 0$ in general. The integers were not assumed at the outset but were a consequence of specifying an initial profile and value for (b/λ_o). Since the physical situation of most interest was the one for which $(b/\lambda_o) >> 1$, it follows that $|\epsilon_\ell/I_\ell| << 1$. The ϵ_ℓ were treated, therefore, as small errors arising from the use of a refractive index profile that deviated from the ideal one, the one that would lead to $\epsilon_\ell=0$.

In the course of the investigation two algorithms were developed for driving the $\epsilon_\ell \to 0$:

$$\text{I.} \qquad \epsilon_\ell \approx \sum_{i=1}^{LL-1} \frac{\partial \epsilon_\ell}{\partial \xi_i} \Delta \xi_i + \ldots \tag{3}$$

$$\text{II.} \qquad \epsilon_\ell \approx \sum_{i=1}^{LL-1} \frac{\partial \epsilon_\ell}{\partial n_i} \Delta n_i + \ldots \tag{4}$$

$$\ell=1,2,\ldots, LL-1$$

The $\Delta\xi_i$ or Δn_i obtained from the inversion of the corresponding system equations was then used for the next iteration, the improved profile being given by

$$\text{I.} \quad \xi_i^{NEW} = \xi_i^{OLD} + \Delta\xi_i \qquad \text{or,} \tag{5}$$

$$\text{II.} \quad n_i^{NEW} = n_i^{OLD} + \Delta n_i \tag{6}$$

The convergance of the Δn_i and $\Delta\xi_i$ was found to be rather slow from one iteration to the next; however, a logarithmic plot of these quantities vs. iteration number revealed a very strong linear variation. It was possible, therefore, to model the asymptotic values of $n_i^{(\infty)}$ and $\xi_i^{(\infty)}$ in closed form using the results from only the first few iterations. The extrapolation worked very well, and for 9 slabs the path difference error between adjacent modes was less than 10^{-5} wavelengths over an axial propagation distance of $.85 \times 10^5$ wavelengths. Other algorithms could have been developed but these two worked sufficiently well, at least for this stage of the study. Numerous tests were conducted to check the accuracy of the eigenvalues and to insure that all of the eigenvalues were being identified in the root-searching process. For 3 slabs surrounded by air, 7-9 significant digits were obtained using single precision arithmetic.

4.3 1 μ_ℓ/Layer Constraints

After some experimentation it was found necessary to satisfy the 1 μ_ℓ/layer constraints before attempting to satisfy the phase constraints. Satisfying both constraints could not be accomplished if the order was reversed.

To handle the 1 μ_ℓ/layer conditions a direct, albeit brute force method was employed. Holding the values of $\{n_i, \xi_i\}$ fixed with $n_1 \equiv 1.50$ and $n_{LL+1} \equiv 1.53$, the corresponding $\{\mu_\ell\}$ were calculated. The same sets of $\{n_i, \xi_i\}$ were used again but this time the ξ_i were scaled up (or down) by a constant factor α, and a new set $\{\mu_\ell\}$ were computed. This process was repeated many times using a DO-loop. In this fashion the correct number of μ's, LL, could be found for at least one of the sets of $\{\xi_i' = \alpha\, \xi_i\}$. Using one (or more) of these sets of ξ_i' the same procedure was repeated on the $\{n_i\}$ except that n_1 was kept fixed at 1.50 and n_{LL+1} at 1.53. In this way, and with a few extra "fine tuning" runs, we were able to obtain a profile $\{n_i', \xi_i'\}$ that at least satisfied the 1 μ_ℓ/layer requirement for as many as 17 slabs.

Using the $\{n_i', \xi_i'\}$ thus obtained for a "starting" profile, $\{n_i^o, \xi_i^o\}$, the program to vary either ξ_i or n_i according to (3) or (4) was then used to attempt to satisfy the phase (or integer) constraints. While we were able to satisfy both the 1 μ_ℓ/layer constraints and the phase constraints for a 9 slab sandwich, we were not able to simultaneously satisfy both for 17 slabs. There appeared to be a problem with the stability of the algorithm, the loss of significant digits, or the nonexistence of solutions that simultaneously satisfy all the constraints for 17 or more slabs. There was insufficient time to see how well the algorithm worked on 13 or 15 slabs.

4.4 Determination of Eigenvalues

The speed and accuracy with which the eigenvalues could be computed were very important factors in this investigation. The analysis developed by Canosa and DeOliveira[(1)] was, therefore, quite attractive for the problem at hand. They analyzed a problem with the boundary conditions $\psi(o) = \psi(L) = 0$, whereas in our problem ψ and ψ' were continuous at all interfaces; consequently, some modifications to their secular equation had to be made.

One of the reasons speed and accuracy were of crucial interest to us is reflected by the manner in which the partial derivatives

$$\frac{\partial \epsilon_\ell}{\partial \xi_i} \quad \text{and} \quad \frac{\partial \epsilon_\ell}{\partial n_i}$$

were evaluated. While these quantities, or the partial deriviatives of the characteristic equation with respect to ξ_i and n_i, could have been found analytically, the task would have been very laborious, and more importantly, prone to errors. To fully appreciate the latter one would have to examine not only the functional form of the characteristic equation as the eigenfunctions composing it changed (for eigenvalues in different layers), but also the programming and debugging ramifications of such an effort while working under deadline constraints. The partial derivatives were, therefore, approximated by finite differences using the same code that computes the eigenvalues. By slightly incrementing one of the parameters of the profile $\{n_i^o, \xi_i^o\}$ at a time, e.g., $\xi_k^o \rightarrow \xi_k^o + \Delta\xi_k$, the altered profile would yield a new set of slightly different eigenvalues and a new set of $\epsilon_\ell = \epsilon_\ell\,(\xi_k^o + \Delta\xi_k)$. The quotients thus obtained, viz.,

$$\frac{\epsilon_\ell(\xi_k + \Delta\xi_k) - \epsilon_\ell(\xi_k)}{\Delta\xi_k} \approx \frac{\partial \epsilon_\ell}{\partial \xi_k} \tag{7}$$

could be and were easily and accurately computed.

4.5 Eigenfunctions and Orthonormalization

With the determination of the eigenvalues completed, the next step was to obtain the normalized amplitudes of the eigenfunctions and to test the orthonormality among the eigenfunctions. Orthonormality between the j^{th} and k^{th} modes requires that

$$N_j^2 \int_{-\infty}^{\infty} \overset{(j)}{\psi}(\xi)\ \overset{(k)}{\psi}(\xi)\ d\xi = \delta_{jk} \tag{8}$$

where the ψ's are the unnormalized eigenfunctions. By setting j=k the N_k for each eigenfunction could be determined. The integral was evaluated analytically in the air regions and numerically in the slabs. For the latter the integration was divided into 3 parts: the middle slab and the two surrounding regions. 48-point Gauss quadrature was used in each of the 3 regions. In this manner all the N's were determined and used to scale the eigenfunctions to the desired normalization.

4.6 Summary and Recommendations

A numerical scheme was devised for obtaining a refractive index profile that would support a set of eigenvalues which satisfied certain design constraints. A variant of the Newton-Raphson method was implemented in the algorithm, as an expedient due to scheduling deadlines. More sophisticated algorithms are available which could be implemented in future work if a larger number of slabs is to be treated in this problem. Branin's[(2)] algorithms in particular should be examined in this regard.

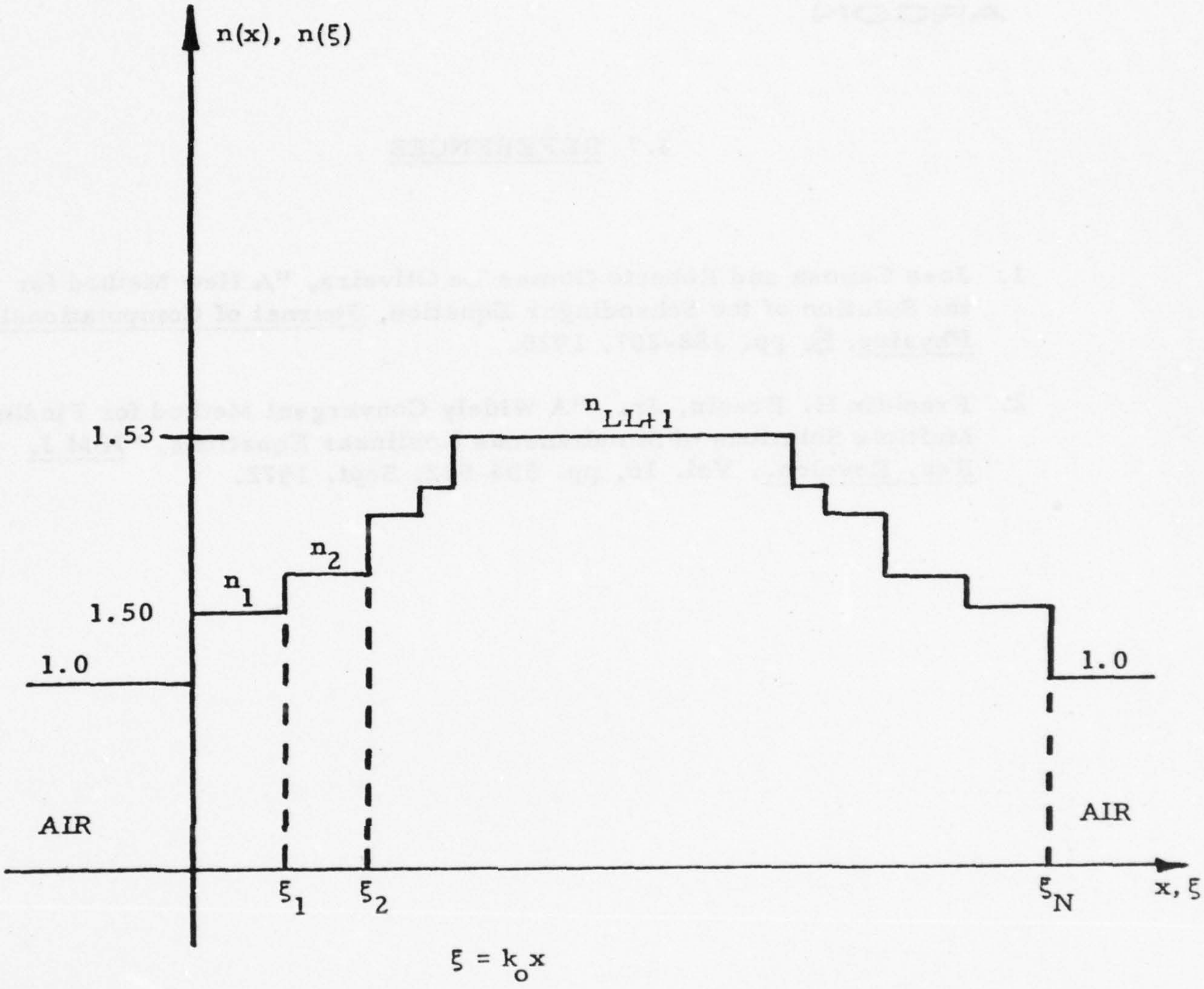


Fig. 4-1

Refractive Index Profile of Multislab

Dielectric Sandwich In Air

4.7 REFERENCES

1. Jose Canosa and Roberto Gomes De Oliveira, "A New Method for the Solution of the Schrodinger Equation, Journal of Computational Physics, **5**, pp. 188-207, 1970.

2. Franklin H. Branin, Jr., "A Widely Convergent Method for Finding Multiple Solutions of Simultaneous Nonlinear Equations," IBM J. Res. Develop., Vol. 16, pp. 504-522, Sept. 1972.

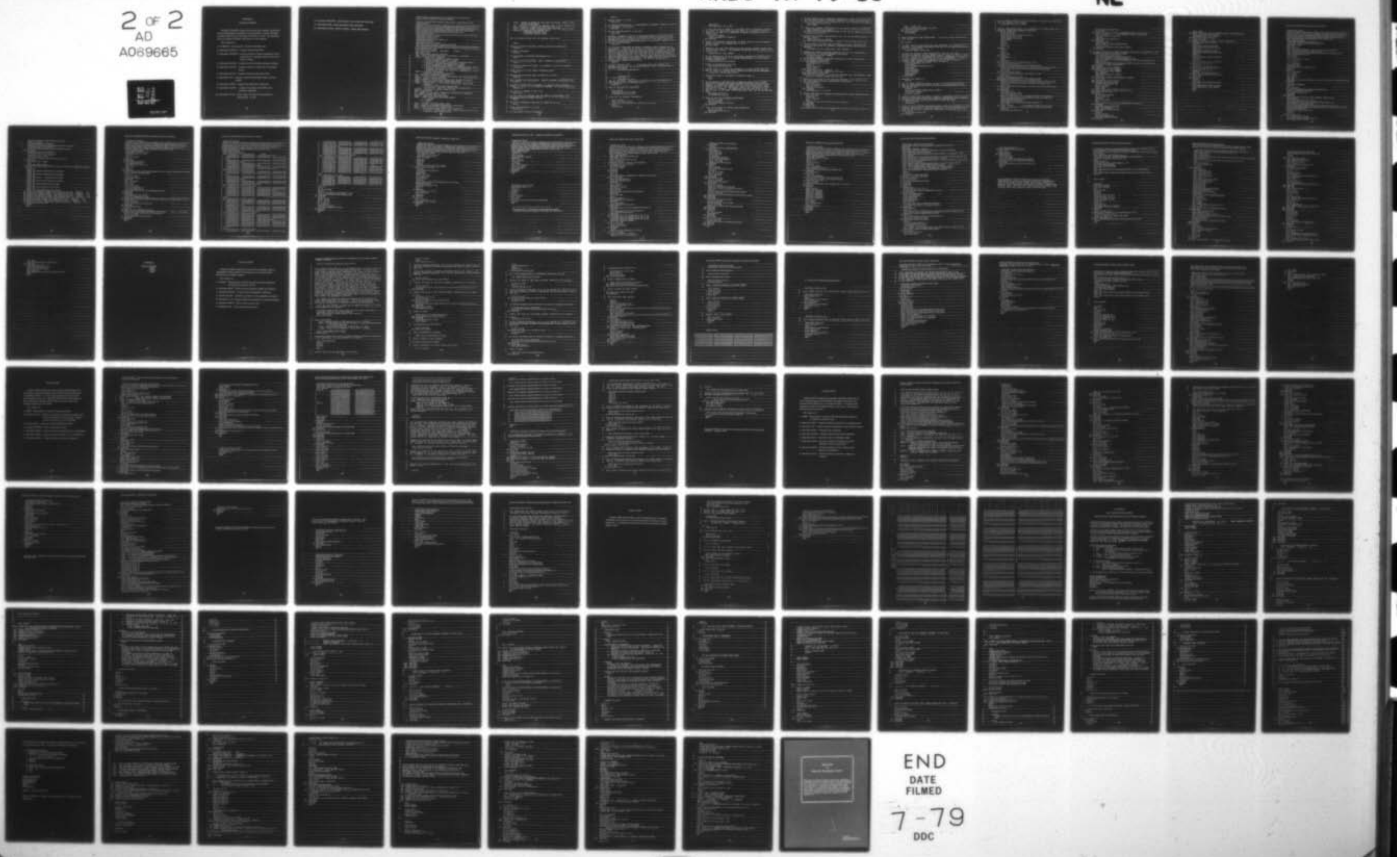
AD-A069 665 ARCON CORP WALTHAM MA F/G 9/2

MATHEMATICAL MODELLING.(U)

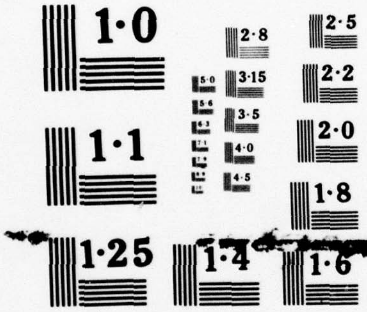
APR 79 S WOOLF, E COHEN F19628-76-C-0247

UNCLASSIFIED RADC-TR-79-65 NL

2 OF 2
AD
A069665



END
DATE
FILMED
7-79
DDC


1·0
2·8
2·5
5·0
3·15
2·2
5·6
3·5
2·0
4·0
1·1
4·5
1·8
1·25
1·4
1·6

APPENDIX 1

Program PLMETHD

Program PLMETHD computes the electron flux Legendre coefficients and associated physical quantities (energy deposition, charge deposition, current profiles) for the scattering of electrons in a slab. The algorithm is the P_ℓ method of solution of the Spencer-Lewis equation.

The routines are

1) PLMETHD - main program, contains integration code

2) Subroutine PROFILE - computes physical quantities

3) Subroutine GOUDSND - performs optional Goudsmit-Saunderson series calculation for a spatially uniform source in an infinite medium.

4) Subroutine BOUNDS - establishes slab (Marshak) boundary conditions

5) Subroutine SCATCO - computes Legendre coefficients of scattering kernel.

6) Subroutine POLYN - computes Legendre polynomial arrays

7) Subroutine LEP - computes Legendre polynomials using recursion formula.

8) Subroutine FZERO - computes the initial flux coefficients.

9) Subroutine LEGEXP - computes flux angular distribution from Legendre coefficients.

10) Subroutine TPLOT - plots results via either Calcomp plotter or printer plots, or both.

11) Subroutine SMLGPLT - plots results on semi-log scale (Calcomp).

12) Subroutine AXUS - draws logarithmic axis (Calcomp).

13) Subroutines PLOT2, PLOT3, PLOT4 - printer plot routines.

MAIN Program- Implements the P_ℓ method for the solution of the Spencer-Lewis equation for electron transport

```
      PROGRAM PLMETHD(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3,TAPE4,TAPE5,
     1TAPE5=OUTPUT)
      COMMON PL(201,51),AMU(201),FSUBL(51,101),FOLD(51,101),PGAUS(6,51),
     XANGMU(6),Q(51),Z(51),SL(51),FOCHRG(101),BACHRG(101),CHRG(101),
     YEDPOS(101),LMAX,NX,LAMBDA,ETA,ISCAT,DS,JSMAX,JSKIP,MUZERO,ISORCE,
     ZKXSRCE,IDBG,IPRNT,METAL,WF(101),WB(101),WNET(101),DELX,ASIG,
     WXSOURCE,EZERO,ECUT,ENRGY,SDRV(51,101),SLC(51,101)
      REAL MUZERO,LAMBDA
      DIMENSION ALPHA(51),BETA(51),AA(5),AQ(5),X(101),F(101),FTOT(101)
      EQUIVALENCE(FTOT(1),F(1))
      DIMENSION DSDX(51,101)
      EQUIVALENCE(FSUBL(1,1),DSDX(1,1))
      DIMENSION FANG(202)
      EQUIVALENCE(FANG(1),WF(1))
      DIMENSION TITLE(3),LMX(5),JSBND(5),DELS(5)
      NAMELIST/PARAMS/MUZERO,XSOURCE,WIDTH,NX,DS,LAMBDA,JSKIP,JSMAX,LMAX
     X,IDBG,ISORCE,ISCAT,ETA,IANGLE,METAL,EZERO,ECUT,ASIG,INTPLT,NORDS,
     YLMX,JSBND,DELS,KALCOMP
      DATA TITLE/10HALUMINUM  ,10HCOPPER    ,10HGOLD      /
      DATA EPS/1.E-6/
      DATA DELS,JSBND,LMX/5*0.,5*0,5*0/
C     IDBG - DEBUG PRINTOUT INDICATOR--0-SUPPRESS,-1-ACTIVATE
C     ISORCE - SOURCE TYPE INDICATOR
C              - 1 SPATIALLY UNIFORM ISOTROPIC SOURCE
C              - 2 PLANE ISOTROPIC SOURCE AT XSOURCE
C              - 3 GAUSSIAN DISTRIBUTED (SPATIALLY) ISOTROPIC SOURCE
C              - 4 SPATIALLY UNIFORM MONODIRECTIONAL SOURCE
C              - 5 PLANE MONODIRECTIONAL SOURCE AT XSOURCE
C              - 6 GAUSSIAN DISTRIBUTED(SPATIALLY) MONODIRECTIONAL SOURCE
C              - 7 LEGENDRE COEFFICIENT OF THE SOURCE FUNCTION ARE
C                TABULATED ON INPUT FILE TAPE5 AS A FUNCTION OF X AND S
C     ISCAT -1 ISOTROPIC SCATTERING
C           - 2 SCREENED RUTHERFORD SCATTERING
C     ETA - RUTHERFORD SCREENING PARAMETER
C     LMAX - (L+1)TH ORDER OF LEGENDRE POLYNOMIALS USED IN CALCULATION
C     MUZERO - SOURCE DIRECTION COSINE
C     JSKIP - NUMBER OF RANGE INTEGRATION STEPS BETWEEN PRINTOUTS
C     JSMAK - TOTAL NUMBER OF RANGE INTEGRATION STEPS
C     IANGLE - ANGULAR FLUX (6 ANGLE GAUSS QUADRATURE) CALCULATION
C              INDICATOR
C              - 1 SUPPRESS ANGULAR FLUX CALCULATION
C              - 2 OR GREATER - ACTIVATE ANGULAR FLUX CALCULATION
C     XSOURCE - POSITION OF SOURCE PLANE IN SLAB
C     WIDTH - SLAB WIDTH
C     NX - NUMBER OF SLAB WIDTH INTEGRATION SUBDIVISION BOUNDARIES
C     DS - RANGE INTEGRATION STEP SIZE
C     LAMBDA - SCATTERING MEAN FREE PATH
C     METAL - INDEX INDICATING SCATTERING MEDIUM
C           - 1 - ALUMINUM
C           - 2 - COPPER
C           - 3 - GOLD
C     EZERO - INITIAL ELECTRON ENERGY (MEV)
C     ASIG - GAUSSIAN DISTRIBUTION PARAMETER (2*VARIANCE)
C     ECUT - LOWER BOUND CUT-OFF ENERGY (MEV)
C     INTPLT - INTERMEDIATE LINE PRINTER PLOT INDICATOR
C              - 0 SUPPRESS INTERMEDIATE PLOTS, SUMMARY PLOT APPEARS ONLY
C              - 1 ACTIVATE INTERMEDIATE PLOTS
```

```
C     NORDS - NUMBER OF VALUES OF LMAX USED IN VARIABLE ORDER P-SUB-L
C             METHOD CALCULATION
C     LMX - VALUES OF LMAX USED IN VARIABLE ORDER P-SUB-L CALCULATION
C     JSBND - NUMBER OF RANGE STEPS FOR EACH LMX VALUE
C     DELS - VALUES OF RANGE INTEGRATION STEP SIZE FOR EACH LMX VALUE
C     KALCOMP - INDICATOR FOR CALCOMP PLOTS
C             - 0 SUPPRESS CALCOMP PLOTS
C             - 1 ACTIVATE CALCOMP PLOTS
C
C
C     SET THE DEFAULT VALUES FOR THE NAMELIST PARAMETERS
C
C
      IDBG=0
C
C     SPATIALLY UNIFORM ISOTROPIC INITIAL ELECTRON DISTRIBUTION
      ISORCE=1
C
C     ISOTROPIC SCATTERING
      ISCAT=1
C
C     IF SCATTERING IS SCREENED RUTHERFORD, ETA = 1.
      ETA=1.
C
C     THIS IS A P-50 CALCULATION.  LMAX = (ORDER OF CALCULATION) + 1
      LMAX=51
C
C     IF SOURCE IS MONODIRECTIONAL, IT IS NORMAL TO THE SLAB SURFACE
      MUZERO=1.0
C
C     PRINT RESULTS AT EVERY RANGE INTEGRATION STEP
      JSKIP=1
C
C     INTEGRATION OVER RANGE WILL BE DONE FOR 10 STEPS
      JSMAX=10
C
C     OMIT ANGULAR FLUX CALCULATION.  COMPUTE LEGENDRE COEFFICIENTS ONLY
      IANGLE=1
C
C     IF INITIAL DISTRIBUTION OR SOURCE IS A SPATIAL DELTA FUNCTION
C     (ISORCE = 2 OR 5), IT IS LOCATED AT XSOURCE = 0.5 RANGE UNITS(R.U.)
      XSOURCE=0.5
C
C     SLAB WIDTH IS ASSUMED TO BE 1 R.U.
      WIDTH=1.0
C
C     THE SLAB WIDTH IS DIVIDED INTO 100 SPATIAL SUBDIVISIONS, (101
C     BOUNDARIES) EACH OF WIDTH .01 R.U.
      NX=101
C
C     THE RANGE INTEGRATION STEP SIZE IS TAKEN AS .01 R.U.
      DS=.01
C
C     TOTAL MEAN-FREE-PATH = 0.1 R.U.
      LAMBDA=0.1
C
C     THE SCATTERING MEDIUM IS ALUMINUM
```

```
      METAL=1
C
C  INITIAL ENERGY = 1.0 MEV
      EZERO=1.0
C
C  IF INITIAL ELECTRON SPATIAL DISTRIBUTION IS GAUSSIAN (ISORCE= 3 OR 6)
C  THE VARIANCE = 0.5/ASIG**2
      ASIG=10.
C
C  CUT-OFF ELECTRON ENERGY = 0.01 MEV
      ECUT=0.01
C
C  SUPPRESS  INDIVIDUAL PLOTS OF CALCULATION RESULTS AFTER EACH RANGE
C  INTEGRATION STEP.  STORE ALL OF THE DATA ON DISK FILE AND MAKE TWO
C  SUMMARY PLOTS, THE ELECTRON FLUX F(X,S) VS. X FOR EACH S VALUE AND
C  THE ENERGY DEPOSITION PROFILE WD(X,S) VS. X FOR EACH S VALUE, AT THE
C  END OF THE CALCULATION.
      INTPLT=0
C
C  THE PROGRAM IS EQUIPPED TO HANDLE A VARIABLE ORDER LEGENDRE SERIES
C  CALCULATION.  THERE CAN BE AS MANY AS FIVE DIFFERENT VALUES OF LMAX,
C  DOWNWARD, LMX(1).GT.LMX(2).GT.LMX(3).GT.LMX(4).GT.LMX(5).  FOR EACH
C  VALUE OF LMX ENTERED, THE USER MUST ALSO ENTER A CORRESPONDING VALUE
C  FOR THE RANGE INTEGRATION STEP SIZE AND THE NUMBER OF SUCH STEPS
C  (DELS(I),JSBND(I),I=1,5).  THIS VARIABLE ORDER FEATURE MAY BE USEFUL
C  WHEN THE SCATTERING IS INITIALLY HIGHLY ANISOTROPIC BUT TENDS TO GROW
C  DIFFUSE AFTER SEVERAL SCATTERINGS.  THE DEFAULT SITUATION IS A
C  SINGLE ORDER CALCUALTION (NORDS=1).
      NORDS=1
C
C  SUPPRESS CALCOMP PLOTS - - - IF CALCOMP PLOTS ARE DESIRED, SET
C  KALCOMP TO 1 AND ADD THE FOLLOWING CONTROL CARDS TO THE DECK BEFORE
C  THE EXECUTION CARD\
C                ATTACH(PEN,ONLINEPEN)
C                       .
C                       .
C                       .
C                LIBRARY(PEN)
C- - - EXECUTION CARD - - -
C                DISPOSE(PLOT,*PL)
C  FOR RED INK PLOTS.  THESE PLOTS ARE SEMILOG PLOTS AND DISPLAY THE
C  SAME QUANTITIES AS THE PRINTER PLOTS.
      KALCOMP=0
C
C  READ IN THE NAMELIST PARAMETERS
C
      READ PARAMS
      IF(NORDS.EQ.1)LMX(1)=LMAX
      IF(NORDS.EQ.1)DELS(1)=DS
C
C  PRINT OUT THE NAMELIST PARAMETERS
C
      PRINT PARAMS
      PRINT 71,TITLE(METAL)
   71 FORMAT(1X,*THE SCATTERING MATERIAL IS*,1X,A10)
      IOPD=1
      LMAX=LMX(1)
      DS=DELS(1)
      IF(NORDS.EQ.1) GO TO 233
C
C  IF MORE THAN ONE PL ORDER IS TO BE USED IN THE CALCULATION, CONVERT
C  THE NUMBER OF RANGE STEPS FOR EACH ORDER TO A CUMULATIVE PARAMETER
C  WHICH WILL BE USED AS AN INDICATOR TO CHANGE THE ORDER OF THE
C  CALCULATION.
      DO 232 I=2,NORDS
      JSBND(I)=JSBND(I-1)+JSBND(I)
  232 CONTINUE
C
C  COMPUTE THE LEGENDRE POLYNOMIALS TO ORDER (LMAX-1) FOR 201 EQUALLY
C  SPACED VALUES OF MU RANGING FROM -1 TO +1
  233 CALL POLYN
C
C  NORMALIZE THE FLUX COEFFICIENTS FOR THE MARSHAK BOUNDARY CONDITIONS
C  (ZERO RETURN CURRENT).  THE BOUNDARY CONDITIONS CAN BE IMPOSED FOR UP
C  TO LMAX = 21.
      IF(LMAX.LE.21) CALL BNDCOF
C
C  IF THE ANGULAR FLUX IS TO BE CALCULATED, COMPUTE THE LEGENDRE POLY-
C  NOMIALS TO ORDER (LMAX-1) AT ANGLES CORRESPONDING TO A SIX POINT
C  GAUSSIAN QUADRATURE.
      IF(IANGLE.NE.1) CALL POLANG
C
C  COMPUTE X-INTEGRATION STEP SIZE
      DELX=WIDTH/(FLOAT(NX-1))
C
C  COMPUTE INDEX IN X-ARRAY CORRESPONDING TO SOURCE CENTER POSITION
C  FOR THE CASE OF A SPATIAL-DELTA FUNCTION OR GAUSSIAN DISTRIBUTION.
      KXSRCE=IFIX( XSOURCE/DELX+EPS)+1
C
C  INITIALIZE RANGE,S, AND RANGE INTEGRATION INDEX JS
      S=0.
      JS=0
C
C  IF THE SOURCE IS OBTAINED FROM A TABULATED FUNCTION OR A KNOWN
C  ANALYTICAL POSITION AND RANGE DEPENDENT FUNCTION, THE ENTRY POINT
C  *SOURCE* IN SUBROUTINE *FZERO* WILL BE CALLED BEFORE EVERY RANGE
C  INTEGRATION STEP.  THE QUANTITIES S AND JS ARE INCREMENTED BEFORE
C  EACH CALL TO *SOURCE*.  IN ORDER THE S=0. AND JS=0 PRIOR TO THE
C  INITIAL CALL IN THE RANGE INTEGRATION LOOP, THEY MUST BE DECREMENTED
C  HERE.
      IF(ISORCE.EQ.7)JS=-1
      IF(ISORCE.EQ.7)S=-DS
C
C  ZERO OUT THE FLUX LEGENDRE COEFFICIENTS
      DO 102 L=1,LMAX
      DO 102 KX=1,NX
      FOLD(L,KX)=FSUBL(L,KX)=0.0
  102 CONTINUE
C
C   FILL IN THE ARRAY OF SPATIAL GRID BOUNDARIES
      DO 120 KX=1,NX
      X(KX)=DELX*FLOAT(KX-1)
  120 CONTINUE
C
```

```
      C  IF THE SOURCE IS NOT A TABULATED FUNCTION OR A KNOWN ANALYTICAL RANGE
      C  AND POSITION DEPENDENT FUNCTION, SUBROUTINE *FZERO* IS CALLED ONLY
      C  ONCE IN THE CALCULATION TO INITIALIZE THE  FLUX LEGENDRE COEFFICIENTS
            IF(ISORCE.EQ.7)GO TO 103
            CALL FZERO
      C
      C   PRINT THE LEGENDRE COEFFICIENTS OF THE INITIAL ANGULAR DISTRIBUTION
            PRINT 720,(SL(L),L=1,LMAX)
        720 FORMAT(/1X,*LEGENDRE COEFFICIENTS OF THE INITIAL ANGULAR DISTRIBUT
           1ION*,/(1X,11E12.5))
      C
      C  COMPUTE THE LEGENDRE COEFFICIENTS, Q(L), OF THE SCATTERING KERNEL
        103 CALL SCATCO
            PRINT 310,(Q(L),L=1,LMAX)
        310 FORMAT(1X,*SCATTERING KERNEL LEGENDRE COEFFICIENTS*/(10E12.5))
      C
      C  IF THE INITIAL ELECTRON FLUX IS SPATIALLY UNIFORM, CALCULATE THE
      C  ELECTRON FLUX USING THE GOUDSMIT SAUNDERSON SERIES FOR CHECKOUT
      C  PURPOSES.
            IF(ISORCE.EQ.1.OR.ISORCE.EQ.4) CALL GOUDSND
      C
      C  COMPUTE THE INITIAL ELECTRON FLUXES AT THE TWO SLAB BOUNDARIES AS
      C  SPECIFIED BY THE MARSHAK BOUNDARY CONDITIONS.
            IF(LMAX.LE.21) CALL BOUNDS
            DO 101 L=1,LMAX
      C  IF EXTENDED PRINTOUT IS DESIRED (IDBG.NE.0), PRINT THE INITIAL
      C  FLUX LEGENDRE COEFFICIENTS.
            IF(IDBG.EQ.0) GO TO 72
            IF(ISORCE.EQ.7) GO TO 72
            LM1=L-1
            PRINT 179,S,LM1
            PRINT 1799
       1799 FORMAT(//2X,8(*X    FSUBL(X)    *))
            PRINT 1798,(X(KX),FSUBL(L,KX),KX=1,NX)
       1798 FORMAT(8(F4.2,F9.5,2X))
        179 FORMAT(///1X,*RANGE = *,E12.5,5X,*LEGENDRE FLUX COEFFICIENTS, ORDE
           1R (L) OF LEGENDRE SERIES TERM IS*,I5)
      C
      C  CALCULATE THE VALUES OF THE COUPLING CONSTANTS ALPHA AND BETA FOR
      C  THE  LMAX COUPLED PARTIAL-DIFFERENTIAL EQUATIONS.
         72 AL=FLOAT(L-1)
            A1=1./(2.*AL+1.)
            ALPHA(L)=AL*A1
            BETA(L)=(1.+AL)*A1
        101 CONTINUE
            IF(ISORCE.EQ.7) GO TO 278
      C
      C  COMPUTE AND PRINT OUT ANGULAR DISTRIBUTION OF INITIAL ELECTRON FLUX
            DO 84 I=1,201
            FANG(I)=0.
            DO 85 L=1,LMAX
            AL=L
            FANG(I)=FANG(I)+0.5*(2.*AL-1.)*SL(L)*PL(I,L)
         85 CONTINUE
         84 CONTINUE
            PRINT 1797
       1797 FORMAT(///1X,*ANGULAR DISTRIBUTION OF INITIAL ELECTRON FLUX*/1X,3(
     1*MU     F(MU)   *))
      PRINT 86,(AMU(I),FANG(I),I=1,201)
   86 FORMAT(8(F5.2,F9.5,1X))
  278 A=DS/(2.*DELX)
      B=DS/LAMBDA
C
C
C     RANGE-INTEGRATION LOOP BEGINS HERE -- JS IS THE RANGE-INTEGRATION
C     STEP COUNTER
C
  200 JS=JS+1
C
C     IF JSMAX RANGE INTEGRATIONS HAVE BEEN PERFORMED, THE CALCULATION IS
C     FINISHED. THE PROGRAM TRANSFERS TO THE END OF THE CODE AND PROCEEDS
C     WITH THE SUMMARY PLOTS.
      IF(JS.GT.JSMAX) GO TO 300
C
C     IF THIS IS A MULTIPLE ORDER CALCULATION, CHECK IF THE RANGE
C     INTEGRATION COUNTER, JS, HAS PASSED A VALUE (JSBND) AT WHICH A CHANGE
C     IN THE PL ORDER AND INTEGRATION STEP SIZE IS INDICATED. IF THIS IS
C     THE CASE, ADJUST THE PARAMETERS DS AND LMAX ACCORDINGLY.  IF THIS IS
C     A SINGLE-ORDER CALCULATION (NORDS=1), THIS SECTION OF CODE IS
C     BYPASSED SINCE IORD WILL ALWAYS HAVE THE VALUE 1.
      IF(IORD.EQ.NORDS) GO TO 30
      IORD1=IORD+1
      IF(JS.LE.JSBND(IORD))GO TO 30
      IORD=IORD1
      DS=DELS(IORD1)
      LMAX=LMX(IORD1)
      A=DS/(2.*DELX)
      B=DS/LAMBDA
   30 CONTINUE
C
C     INCREMENT THE RANGE, S.
      S=S+DS
C
C     SET THE PRINT PARAMETER, IPRNT, SO THAT THE INTERMEDIATE RESULTS
C     WILL BE PRINTED AND PLOTTED FOR THE FIRST AND EVERY JSKIP-TH RANGE
C     INTEGRATION STEP.
      IPRNT=0
      IF((JS-(JS/JSKIP)*JSKIP).EQ.0) IPRNT=1
      IF(JS.EQ.1) IPRNT=1
      DO 202 L=1,LMAX
C
C     CLEAR THE ARRAY FTOT(KX) WHICH IS USED TO TEMPORARILY STORE THE TOTAL
C     ELECTRON FLUX(ZEROTH ORDER FLUX LEGENDRE COEFFICIENT) AS A FUNCTION
C     OF POSITION IN THE SLAB AT EACH RANGE INTEGRATION STEP.
      DO 202 KX=1,NX
      FTOT(KX)=0.
C
C     STORE THE FLUX LEGENDRE COEFFICIENT ARRAY FSUBL(L,KX) COMPUTED AT THE
C     PREVIOUS RANGE STEP IN THE ARRAY FOLD(L,KX) SO THAT THE ARRAY FSUBL
C     WILL NOT BE DESTROYED DURING THE SUBSEQUENT COMPUTATION OF THE NEW
C     FSUBL ARRAY.
      FOLD(L,KX)=FSUBL(L,KX)
  202 CONTINUE
C
```

```
C  IF THE SOURCE FUNCTION IS RANGE DEPENDENT (TABULATED OR ANALYTIC),
C  CALL THE *SOURCE* ENTRY POINT.
       IF(ISORCE.EQ.7)CALL SOURCE
C
C
C  SPATIAL INTEGRATION LOOP.  THE ALGORITHM IS A SECOND ORDER LAX
C  WENDROFF SCHEME (SEE SEPARATE WRITE-UP).
       DO 201 L=1,LMAX
       DO 201 KX=1,NX
       AA(1)=AA(2)=AA(3)=AA(4)=AA(5)=0.0
       IF(KX.EQ.1) GO TO 400
       IF(KX.EQ.NX) GO TO 410
       KK=KP=KX+1
       KM=KX
       KQ=KX-1
       SYN=0.5
       GO TO 420
   400 KK=3
       KM=KP=2
       KQ=1
       SYN=1.0
       GO TO 420
   410 KK=NX-2
       KM=KP=NX-1
       KQ=NX
       SYN=-1.0
   420 IF(L.GT.2.AND.L.LT.(LMAX-1))GO TO 444
       IF(L.GT.1) GO TO 441
   419 AA(2)=2.*A*SYN*(FOLD(L+1,KP)-FOLD(L+1,KQ))
       AA(4)=2.*A**2*(FOLD(L+2,KK)-2.*FOLD(L+2,KM)+FOLD(L+2,KQ))
       AA(5)=2.*A**2*(FOLD(L,KK)-2.*FOLD(L,KM)+FOLD(L,KQ))
       GO TO 445
   441 IF(L.GT.2) GO TO 442
       AA(1)=2.*A*SYN*(FOLD(L-1,KP)-FOLD(L-1,KQ))
       GO TO 419
   442 IF(L.EQ.LMAX) GO TO 443
   446 AA(2)=2.*A*SYN*(FOLD(L+1,KP)-FOLD(L+1,KQ))
   443 AA(1)=2.*A*SYN*(FOLD(L-1,KP)-FOLD(L-1,KQ))
       AA(3)=2.*A**2*(FOLD(L-2,KK)-2.*FOLD(L-2,KM)+FOLD(L-2,KQ))
       AA(5)=2.*A**2*(FOLD(L,KK)-2.*FOLD(L,KM)+FOLD(L,KQ))
       GO TO 445
   444 AA(4)=2.*A**2*(FOLD(L+2,KK)-2.*FOLD(L+2,KM)+FOLD(L+2,KQ))
       GO TO 446
   445 CONTINUE
       BB=3*(Q(L)-1.)
       IF(L.EQ.1) GO TO 450
       IF(L.EQ.LMAX)GO TO 460
       AQ(1)=-(((Q(L)+Q(L-1)-2.)/LAMBDA*0.5*DS)+1.0)*ALPHA(L)
       AQ(2)=-(((Q(L)+Q(L+1)-2.)/LAMBDA*0.5*DS)+1.0)*BETA(L)
       AQ(3)=ALPHA(L)*ALPHA(L-1)
       AQ(4)=BETA(L)*BETA(L+1)
       AQ(5)=ALPHA(L+1)*BETA(L)+ALPHA(L)*BETA(L-1)
       GO TO 465
   450 AQ(1)=-(((Q(L)          -2.)/LAMBDA*0.5*DS)+1.0)*ALPHA(L)
       AQ(2)=-(((Q(L)+Q(L+1)-2.)/LAMBDA*0.5*DS)+1.0)*BETA(L)
       AQ(3)=0.
       AQ(4)=BETA(L)*BETA(L+1)
```

```
      AQ(5)=ALPHA(L+1)*BETA(L)
      GO TO 465
  460 AQ(1)=-(((Q(L)+Q(L-1)-2.)/LAMBDA*0.5*DS)+1.0)*ALPHA(L)
      AQ(2)=-(((Q(L)          -2.)/LAMBDA*0.5*DS)+1.0)*BETA(L)
      AQ(3)=ALPHA(L)*ALPHA(L-1)
      AQ(4)=0.
      AQ(5)=ALPHA(L)*BETA(L-1)
  465 CONTINUE
      FAC=FOLD(L,KX)
      FSUBL(L,KX)=(1.+BB+0.5*BB**2)*FAC
      DO 150 IT=1,5
      FSUBL(L,KX)=FSUBL(L,KX)+AA(IT)*AQ(IT)
  150 CONTINUE
C
C  IF SOURCE FUNCTION IS RANGE DEPENDENT(TABULATED OF ANALYTICAL), ADD
C  APPROPRIATE SOURCE DERIVATIVE TERMS.
      IF(ISORCE.NE.7) GO TO 201
      IF(L.EQ.1)DSTERM=BETA(L)*DSDX(L+1,KX)
      IF(L.EQ.LMAX)DSTERM=ALPHA(L)*DSDX(L-1,KX)
      IF(L.GT.1.AND.L.LT.LMAX)DSTERM=BETA(L)*DSDX(L+1,KX)+ALPHA(L)*
     1DSDX(L-1,KX)
      FSUBL(L,KX)=FSUBL(L,KX)+SLC(L,KX)*DS*(1.+(Q(L)-1.)*DS/LAMBDA)+
     10.5*DS**2*(SDRV(L,KX)-DSTERM)
  201 CONTINUE
  703 FORMAT(/1X,*L=*,I5,*FOLD*)
  704 FORMAT(/1X,*L=*,I5,*FSUBL*)
      IF(LMAX.LE.21) CALL BOUNDS
      CALL PROFILE
      IF(ENRGY.GT.0.0) GO TO 188
      PRINT 187
  187 FORMAT(/1X,*ENERGY BELOW CUT-OFF VALUE*)
      GO TO 300
  188 IF(IPRNT.NE.1) GO TO 200
  189 FORMAT(//1X,*S=*,E12.5)
      IF(IDBG.NE.1) GO TO 215
      DO 211 L=1,LMAX
      PRINT 179,S,L
      PRINT 1799
  211 PRINT 1798,(X(KX),FSUBL(L,KX),KX=1,NX)
  180 FORMAT(1X,10E13.5)
  215 KPRNT=0
      DO 213 NMU=1,201,20
      DO 212 KX=1,NX
      F(KX)=0.
      DO 212 L=1,LMAX
      AL=L
      F(KX)=F(KX)+0.5*(2.*AL-1.)*FSUBL(L,KX)*PL(NMU,L)
  212 CONTINUE
      IF(KPRNT.EQ.0) GO TO 213
      PRINT 214,AMU(NMU)
      PRINT 180,(F(KX),KX=1,NX)
  213 CONTINUE
      FINT=0.
      DO 229 KX=1,NX
      FTOT(KX)=FSUBL(1,KX)
      FINT=FTOT(KX)*DELX+FINT
  229 CONTINUE
```

```
      PRINT 230,S
  230 FORMAT(//1X,*S=*,E12.5,*CHARGE SPATIAL DISTRIBUTION*)
      PRINT 180,(FTOT(KX),KX=1,NX)
      IF(IANGLE.NE.1) CALL LEGEXP
  214 FORMAT(/1X,*MU=*,E12.5)
      PRINT 242,FINT
  242 FORMAT(/1X,*TOTAL CHARGE IN SLAB =*,2X,E16.9)
      IF(INTPLT.EQ.0) GO TO 60
      REWIND 3
      WRITE(3)JS,(X(KX),KX=1,NX)
      WRITE(3)JS,(FTOT(KX),KX=1,NX)
      REWIND 4
      WRITE(4)JS,(X(KX),KX=1,NX)
      WRITE(4)JS,(EDPOS(KX),KX=1,NX)
      REWIND 3
      JPLOT=3
      CALL TPLOT(NX,6,JPLOT,KALCOMP)
      REWIND 4
      JPLOT=4
      CALL TPLOT(NX,6,JPLOT,KALCOMP)
   60 IF(JS.GT.1) GO TO 61
      REWIND 1
      WRITE(1)JS,(X(KX),KX=1,NX)
   61 WRITE(1)JS,(FTOT(KX),KX=1,NX)
      IF(JS.GT.1) GO TO 62
      REWIND 2
      WRITE(2) JS,(X(KX),KX=1,NX)
   62 WRITE(2) JS,(EDPOS(KX),KX=1,NX)
      GO TO 200
  300 REWIND 1
      REWIND 2
      JPLOT=1
      CALL TPLOT(NX,6,JPLOT,KALCOMP)
      JPLOT=2
      CALL TPLOT(NX,6,JPLOT,KALCOMP)
      IF(KALCOMP.EQ.1) CALL ENDPLT
      STOP
      END
```

Subroutine PROFILE-computes physical quantities

```
      SUBRJUTINE PROFILE
      COMMON PL(201,51),AMU(201),FSUBL(51,101),FOLD(51,101),PGAUS(6,51),
     XANGMJ(6),O(51),Z(51),SL(51),FOCHRG(101),BACHRG(101),CHPG(101),
     YEDPOS(101),LMAX,NX,LAMBDA,ETA,ISCAT,DS,JSMAX,JSKIP,MUZERO,ISORCE,
     ZKXSRCE,IDBG,IPRNT,METAL,WF(101),WB(101),WNET(101),DELX,ASIG,
     WXSOJRCE,EZERO,ECJT,ENRGY,SDRV(51,101),SLC(51,101)
      REAL MUZEPO,LAMBDA
      DIMENSION FRSTMOM(51),FCTRL(51)
      DIMENSION ZZ(3),AZ(3),EXCIT(3),DNSTY(3)
      DIMENSION X(101)
      REAL MC2
      DATA ITIME/0/
      DATA ZZ,AZ,EXCIT,DNSTY/13.,29.,79.,26.98,63.54,197.0,
     X163.E-6,80.E-6,7.97E-6,2.7,8.9,10.5/
      DATA RE,AVOG,TWOPI,MC2/.2818E-12,6.025E23,6.2831853,0.511/
      IF(ITIME.GT.0)GO TO 100
      ENRGY=FZERO
      YL2=ALOG(2.)
      FCC=TWOPI*AVOG*RE**2*MC2
      FCTRL(1)=1.
      FRSTMOM(1)=0.5
      DO 1 L=2,LMAX
      FRSTMOM(L)=0.0
      FCTRL(L)=FLOAT(L-1)*FCTRL(L-1)
    1 CONTINUE
      SGN=-1.
      DO 2 L=3,LMAX,2
      LL=(L+3)/2
      LM=(L-1)/2
      AL=FLOAT(L-1)
      SGN=-SGN
      FRSTMOM(L)=(2.*AL+1.)*FCTRL(L-2)/((2.**L)*FCTRL(LL)*FCTRL(LM))*SGN
    2 CONTINUE
      ITIME=1
C     INITIALIZE PROFILES
      X(1)=0.
      DO 3 KX=1,NX
      FOCHRG(KX)=BACHRG(KX)=CHRG(KX)=EDPOS(KX)=0.
      WF(KX)=WB(KX)=WNET(KX)=0.0
      X(KX)=DELX*FLOAT(KX-1)
    3 CONTINUE
  100 CONTINUE
C     BERGER-SELTZER STOPPING POWER FORMULA COMPUTATION
      TK=ENRGY/MC2
      BETA=(SQRT(TK*(TK+2.)))/(TK+1.)
      XLG=ALOG(TK**2*(TK+2.)/(2.*(EXCIT(METAL)/MC2)**2))
C     RELATIVISTIC CORRECTION TERM
      FM=1.-BETA**2+(0.125*TK**2-(2.*TK+1.)*XL2)/(TK+1.)**2
      DEBYDS=-(FCC*DNSTY(METAL)*ZZ(METAL)/AZ(METAL)/BETA**2)*(XLG+FM)
      ENRGY=ENRGY+DEBYDS*DS
      IF(ENRGY.GT.ECUT) GO TO 112
      ENRGY=-5.
      RETURN
  112 DO 102 KX=1,NX
C     NET CHARGE FLOW PROFILE
      CHRG(KX)=CHRG(KX)+FSUBL(2,KX)*DS
C     NET ENERGY FLOW PROFILE
```

```
      WNET(KX)=WNET(KX)+ENRGY*FSUBL(2,KX)*DS
      DO 101 L=1,LMAX
C     FORWARD CHARGE FLOW PROFILE
      FOCHRG(KX)=FOCHRG(KX)+FRSTMOM(L)*FSUBL(L,KX)*DS
C     FORWARD ENERGY FLOW PROFILE
      WF(KX)=WF(KX)+FRSTMOM(L)*ENRGY*FSUBL(L,KX)*DS
  101 CONTINUE
C     BACKWARD CHARGE FLOW PROFILE
      BACHRG(KX)=CHRG(KX)-FOCHRG(KX)
C     BACKWARD ENERGY FOLW PROFILE
      WB(KX)=WNET(KX)-WF(KX)
      EDPOS(KX)=EDPOS(KX)-DEBYDS*FSUBL(1,KX)*DS
  102 CONTINUE
      IF(IPRNT.NE.1) RETURN
      NN=1
      PRINT 60,ENRGY,DEBYDS
   60 FORMAT(/1X,*ELECTRON ENERGY=*,E12.5,1X,*MEV*/1X,*STOPPING POWER=*,
     1E12.5)
      PRINT 50
      PRINT 40,(X(KX),FOCHRG(KX),KX=1,NX,NN)
      PRINT 51
      PRINT 40,(X(KX),BACHRG(KX),KX=1,NX,NN)
      PRINT 52
      PRINT 40,(X(KX),  CHRG(KX),KX=1,NX,NN)
      PRINT 53
      PRINT 40,(X(KX),EDPOS (KX),KX=1,NX,NN)
      PRINT 54
      PRINT 40,(X(KX),WF(KX),KX=1,NX,NN)
      PRINT 55
      PRINT 40,(X(KX),WB(KX),KX=1,NX,NN)
      PRINT 56
      PRINT 40,(X(KX),WNET(KX),KX=1,NX,NN)
   50 FORMAT(/1X,*FORWARD CHARGE FLOW PROFILE*/2X,8(*X    PROFILE     *))
   54 FORMAT(/1X,*FORWARD ENERGY FLOW PROFILE*/2X,8(*X    PROFILE     *))
   51 FORMAT(/1X,*BACKWARD CHARGE FLOW PROFILE*/2X,8(*X    PROFILE     *))
   55 FORMAT(/1X,*BACKWARD ENERGY FLOW PROFILE*/2X,8(*X    PROFILE     *))
   56 FORMAT(/1X,*NET ENERGY FLOW PROFILE*/2X,8(*X    PROFILE     *))
   52 FORMAT(/1X,*NET CHARGE FLOW PROFILE*/2X,8(*X    PROFILE     *))
   53 FORMAT(/1X,*ENERGY DEPOSITION PROFILE*/2X,8(*X    PROFILE     *))
   40 FORMAT(8(F4.2,F9.5,2X))
      RETURN
      END
```

Subroutine GOUDSND-Goudsmit-Saunderson series calculation

```
      SUBROUTINE GOUDSND
      COMMON PL(201,51),AMU(201),FSUBL(51,101),FOLD(51,101),PGAUS(6,51),
     XANGMU(6),Q(51),Z(51),SL(51),FOCHRG(101),BACHRG(101),CHRG(101),
     YEDPOS(101),LMAX,NK,LAMBDA,ETA,ISCAT,DS,JSMAX,JSKIP,MUZERO,ISORCE,
     ZKXSRCE,IDBG,IPRNT,METAL,WF(101),WB(101),WNET(101),DELX,ASIG,
     WXSOURCE,EZERO,ECUT,ENRGY,SDRV(51,101),SLC(51,101)
      REAL MUZERO,LAMBDA
      DIMENSION S(101),F(6,11)
      IF(ISCAT.NE.1) GO TO 1
      Z(1)=0.
      DO 10 L=2,LMAX
      Z(L)=1.
   10 CONTINUE
    1 DELS=DS*FLOAT(JSKIP)
      JMAX=JSMAX/JSKIP+1
      IF(JMAX.LE.11) GO TO 11
      PRINT 12
      STOP
   11 CONTINUE
   12 FORMAT(1X,*JSMAX AND JSKIP VALUES ARE NOT USABLE IN GOUDSMIT-SAUND
     XERSON SERIES CALCULATION*)
      S(1)=0.
      DO 2 J=2,JMAX
      S(J)=S(J-1)+DELS
    2 CONTINUE
      DO 4 J=1,JMAX
      DO 4 K=1,6
      F(K,J)=0.
      DO 40 L=1,LMAX
      AL=0.5*FLOAT(2*L-1)
      F1=S(J)/LAMBDA*Z(L)
      F(K,J)=F(K,J)+AL*EXP(-F1)*PGAUS(K,L)*SL(L)
   40 CONTINUE
    4 CONTINUE
      IF(ISCAT.NE.1) GO TO 60
      PRINT 50, (ANGMU(I),I=1,6)
   50 FORMAT(1X,*ISOTROPIC SCATTERING*/9X,*COSINE OF POLAR ANGLE*/7X,*S*
     X,5X,6E13.6)
      DO 51 J=1,JMAX
      PRINT 52,S(J),(F(K,J),K=1,6)
   51 CONTINUE
   52 FORMAT(1X,7E13.6)
      RETURN
   60 PRINT 53,ETA,(ANGMU(I),I=1,6)
   53 FORMAT(//1X,*SCREENED RUTHERFORD SCATTERING - - ETA =*, E12.5/9X,*
     XCOSINE OF POLAR ANGLE*/7X,*S*,5X,6E13.6)
      DO 54 J=1,JMAX
      PRINT 52,S(J),(F(K,J),K=1,6)
   54 CONTINUE
      RETURN
      END
```

Subroutine BOUNDS-Marshak boundary conditions

```
      SUBROUTINE BOUNDS
      COMMON PL(201,51),AMU(201),FSUBL(51,101),FOLD(51,101),PSAUS(6,51),
     XANGMJ(6),Q(51),Z(51),SL(51),FOCHRG(101),BACHRG(101),CHRG(101),
     YEDPOS(101),LMAX,NX,LAMBDA,ETA,ISCAT,DS,JSMAX,JSKIP,MUZERO,ISORCE,
     ZKXSRCF,IDBG,IPRNT,METAL,WF(101),WB(101),WNET(101),DELX,ASIG,
     WXSOURCE,EZERO,ECUT,ENRGY,SDRV(51,101),SLC(51,101)
      REAL MUZERO,LAMBDA
      DIMENSION COEFS(21,21)
      DATA ((COEFS(I,J),I=1,21),J=1,7)/       42*0.,
     1   .100000000E+01, .108253174E+01, .447213573E+00,18*0.,
     2   .100000000E+01, .121243547E+01, .715541553E+00, .193431191E+00,
     3  17*0.,
     4   .100000000E+01, .129903767E+01, .912679808E+00, .389774887E+00,
     5   .816322106E-01,16*0.,
     6   .100000000E+01, .136089570E+01, .106479092E+01, .563005717E+00,
     7   .190473719E+00, .320796799E-01,15*0.,
     8   .100000000E+01, .140728789E+01, .118578474E+01, .716546279E+00,
     9   .308531086E+00, .867588344E-01, .122238148E-01,14*0./
      DATA((COEFS(I,J),I=1,21),J=8,10)/
     1   .100000000E+01, .144336841E+01, .128429581E+01, .851324245E+00,
     2   .427184791E+00, .156056330E+00, .376093446E-01, .455527160E-02,
     3  13*0.,
     4   .100000000E+01, .147222939E+01, .136599506E+01, .971053492E+00,
     5   .542171113E+00, .234067819E+00, .744588241E-01, .157134773E-01,
     6   .166907415E-02,12*0.,
     7   .100000000E+01, .149583829E+01, .143484043E+01, .107646412E+01,
     8   .651472159E+00, .316646843E+00, .120429966E+00, .339754287E-01,
     9   .638003544E-02, .603426106E-03,11*0./
      DATA ((COEFS(I,J),I=1,21),J=11,12)/
     1   .100000000E+01, .151550699E+01, .149361738E+01, .117025674E+01,
     2   .754267233E+00, .400960565E+00, .173246609E+00, .589939019E-01,
     3   .149638848E-01, .253166368E-02, .215782949E-03,10*0.,
     4   .100000000E+01, .153214456E+01, .154437097E+01, .125355082E+01,
     5   .850381728E+00, .485115271E+00, .230940864E+00, .909144615E-01,
     6   .277784546E-01, .640327792E-02, .985803104E-03, .764587205E-04,
     7  9*0./
      DATA ((COEFS(I,J),I=1,21),J=13,14)/
     1   .100000000E+01, .154640280E+01, .158863449E+01, .132340187E+01,
     2   .939989604E+00, .567879422E+00, .291908455E+00, .126135796E+00,
     3   .448407219E-01, .126606545E-01, .267523227E-02, .377355143E-03,
     4   .268830186E-04,8*0.,
     5   .100000000E+01, .155875876E+01, .162758287E+01, .139583606E+01,
     6   .102344883E+01, .648487307E+00, .354891106E+00, .166455790E+00,
     7   .659509115E-01, .215692874E-01, .561522506E-02, .109546469E-02,
     8   .142930672E-03, .939186884E-05,7*0./
      DATA ((COEFS(I,J),I=1,21),J=15,16)/
     1   .100000000E+01, .156957566E+01, .166213104E+01, .145589219E+01,
     2   .110120153E+01, .726493597E+00, .418927491E+00, .210144877E+00,
     3   .907876939E-01, .332465544E-01, .100741908E-01, .243386347E-02,
     4   .441065058E-03, .534763094E-04, .326454756E-05,6*0.,
     5   .100000000E+01, .157912776E+01, .169299638E+01, .151242519E+01,
     6   .117370319E+01, .801661253E+00, .483289957E+00, .256469965E+00,
     7   .118971566E+00, .476967096E-01, .162433385E-01, .459309571E-02,
     8   .103458762E-02, .175083997E-03, .198277412E-04, .113049353E-05,
     9  5*0./
      DATA ((COEFS(I,J),I=1,21),J=17,18)/
     1   .100000000E+01, .158764505E+01, .172090135E+01, .156325340E+01,
     2    .124153042E+01, .874030113E+00, .547568321E+00, .304935401E+00,
     3    .150167942E+00, .648779869E-01, .242874622E-01, .774139165E-02,
     4    .204823911E-02, .433061272E-03, .687928287E-04, .730933447E-05,
     5    .391068170E-06,4*0.,
     6    .100000000E+01, .159519196E+01, .174564362E+01, .160330634E+01,
     7    .130416870E+01, .942504883E+00, .610458374E+00, .354164124E+00,
     8    .183532715E+00, .844192505E-01, .341224670E-01, .119319641E-01,
     9    .358533359E-02, .892758369E-03, .177308917E-03, .264868140E-04,
     1    .264868140E-05, .133266440E-06,3*0./
      DATA ((COEFS(I,J),I=1,21),J=19,20)/
     1    .100000000E+01, .160228729E+01, .176956940E+01, .165489197E+01,
     2    .136679077E+01, .101199341E+01, .676849365E+00, .487241921E+00,
     3    .221191406E+00, .107452393E+00, .464744569E-01, .177569525E-01,
     4    .590515137E-02, .167393684E-02, .390410423E-03, .740736723E-04,
     5    .106506050E-04, .101327896E-05, .486033369E-07,   2*0.        ,
     6    .100000000E+01, .160498047E+01, .177563477E+01, .166345703E+01,
     7    .138378906E+01, .104980469E+01, .696289063E+00, .420410156E+00,
     8    .237304688E+00, .115966797E+00, .512695313E-01, .205688477E-01,
     9    .577490234E-02, .229644775E-02, .514964131E-03, .107765193E-03,
     1    .190734663E-04, .196695328E-05, .130385160E-05, .769341124E-08,
     2 0./
      DATA (COEFS(I,21),I=1,21)/
     1    .100000000E+01, .166038513E+01, .197094727E+01, .206768799E+01,
     2    .185338184E+01, .155871582E+01, .124914551E+01, .357055664E+00,
     3    .669372559E+00, .325561523E+00, .185394287E+00, .978240967E-01,
     4    .404968262E-01, .160140991E-01, .577831268E-02, .177383423E-02,
     5    .421404839E-03, .702738762E-04, .112950802E-04, .810250640E-06,
     5    .499931048E-07 /
      LM=LMAX
      AA=1.
      DO 20 L=2,LMAX
      AA=-AA
      FSUBL(L,1)=COEFS(L,LM)*FSUBL(1,1)*AA
      FSUBL(L,NX)=COEFS(L,LM)*FSUBL(1,NX)
   20 CONTINUE
      RETURN
      ENTRY BNDCOF
      DO 10 J=1,21
      DO 10 I=1,21
      AI=FLOAT(I)
      FAC=SQRT(2.*AI-1.)
      COEFS(I,J)=COEFS(I,J)/FAC
   10 CONTINUE
      RETURN
      END
```

Subroutine SCATCO-Legendre scattering coefficients

```
      SUBROUTINE SCATCO
      COMMON PL(201,51),AMU(201),FSUBL(51,101),FOLD(51,101),PGAUS(6,51),
     XANGMU(6),Q(51),Z(51),SL(51),FOCHPG(101),BACHPG(101),CHRG(101),
     YEDPOS(101),LMAX,NX,LAMBDA,ETA,ISCAT,DS,JSMAX,JSKIP,MUZERO,ISOFCE,
     ZKXSRCE,IDBG,IPRNT,METAL,WF(101),WB(101),WNET(101),DELX,ASIG,
     WXSOURCE,EZERO,ECUT,ENRGY,SORV(51,101),SLC(51,101)
      REAL MUZERO,LAMBDA
      IF(ISCAT.GT.1) GO TO 10
C     ISOTROPIC SCATTERING
      DO 5 L=1,LMAX
      Q(L)=0.
    5 CONTINUE
      Q(1)=1.
      RETURN
C     SCREENED RUTHERFORD SCATTERING
   10 E1=(1./(1.+0.5*ETA)   )
      Z(1)=0.
      Z(2)=ALOG(1.+2./ETA)-E1
      C1=E1/ETA
      DMIN=1.E6
      DO 11 L=3,LMAX
      AL=FLOAT(L-2)
      BL=2.*AL+1.
      CL=AL+1.
      Z(L)=(BL*(1.+ETA)*Z(L-1)-CL*Z(L-2)-BL*E1)/AL
      DD=ABS(Z(L)-C1)
      IF(DD.GT.DMIN)GO TO 11
      DMIN=DD
      LM=L
   11 CONTINUE
      IF(LM.EQ.51) GO TO 17
      DO 18 L=LM,51
      Z(L)=C1
   18 CONTINUE
   17 Q(1)=1.
      DO 12 L=1,51
      Q(L)=Q(1)-ETA/E1*Z(L)
   12 CONTINUE
      RETURN
      END
```

Subroutines POLYN, LEP - Legendre polynomial computation

```
      SUBR3UTINE POLYN
      COMMON PL(201,51),AMU(201),FSUBL(51,101),FOLD(51,101),PGAUS(5,51),
     XANGMU(6),Q(51),Z(51),SL(51),FOCHRG(101),3ACHRG(101),CHRG(101),
     YEDPOS(101),LMAX,NX,LAMBDA,ETA,ISCAT,DS,JSMAX,JSKIP,MUZERO,ISORCE,
     ZKXSRCF,IDBG,IPRNT,METAL,WF(101),WB(101),WNET(101),DELX,ASIG,
     WXSOURCE,EZERO,EOUT,ENRGY,SDRV(51,101),SLC(51,101)
      REAL MUZERO,LAMBDA
      DIMENSION Y(51)
      AMU(1)=-1.
      DO 1 I=2,201
      AMU(I)=AMU(I-1)+0.01
    1 CONTINUE
      DO 2 I=1,201
      X=AMU(I)
      LL=LMAX-1
      CALL LEP(Y,X,LL)
      DO 2 L=1,LMAX
      PL(I,L)=Y(L)
    2 CONTINUE
      RETURN
      END
```

*

```
      SUBROUTINE LEP(Y,X,N)
      DIMENSION Y(51)
      Y(1)=1.
      IF(N)1,1,2
    1 RETURN
    2 Y(2)=X
      IF(N-1)1,1,3
    3 DO 4 I=2,N
      G=X*Y(I)
    4 Y(I+1)=G-Y(I-1)+G-(G-Y(I-1))/FLOAT(I)
      RETURN
      END
```

* Subroutine LEP is extracted from the IBM System/360 Scientific Soubroutine Package, Publication N0. H20-0205-3

Subroutine FZERO-Initial flux coefficients

```
      SUBROUTINE FZERO
      COMMON PL(201,51),AMU(201),FSUBL(51,101),FOLD(51,101),PGAUS(5,51),
     XANGMU(6),Q(51),Z(51),SL(51),FOCHRG(101),BACHRG(101),CHRG(101),
     YEDPOS(101),LMAX,NX,LAMBDA,ETA,ISCAT,DS,JSMAX,JSKIP,MUZERO,ISORCE,
     ZKXSRCE,IDBG,IPRNT,METAL,WF(101),WB(101),WNET(101),DELX,ASTG,
     WXSOURCE,EZERO,EOUT,ENRGY,SDRV(51,101),SLC(51,101)
      REAL MUZERO,LAMBDA
      DIMENSION SUM(201),SIGMA(100)
      DATA EPS/1.E-6/
      DATA PI,H/3.141592654,0.01/
      DATA ISTART/0/
C     DETERMINE INITIAL ANGULAR DISTRIBUTION
      IF(ISORCE.GT.3) GO TO 20
C     ISOTROPIC SOURCE
      DO 10 L=1,LMAX
      SL(L)=0.
   10 CONTINUE
      SL(1)=1.
      GO TO 30
C     MONODIRECTIONAL SOURCE(DELTA FUNCTION IN ANGLE)
   20 DO 1 N=1,100
      ARG=PI*FLOAT(N)/100.
      SIGMA(N)=SIN(ARG)/ARG
    1 CONTINUE
      DO 2 I=1,201
      SUM(I)=0.5
      DIF=AMU(I)-MUZERO
      DO 2 N=1,99
      AN=FLOAT(N)
      SUM(I)=SUM(I)+SIGMA(N)*COS(AN*DIF)
    2 CONTINUE
      DO 3 L=1,LMAX
      ANS=0.
      DO 4 I=2,200,2
      ANS=ANS+SUM(I)*PL(I,L)*4.
    4 CONTINUE
      DO 5 I=3,199,2
      ANS=ANS+SUM(I)*PL(I,L)*2.
    5 CONTINUE
      ANS=ANS+SUM(1)*PL(1,L)+SUM(201)*PL(201,L)
      FAC=1.0
      IF((ABS(MUZERO-1.0)).LT.EPS)FAC=2.
      SL(L)=ANS*H*FAC/(3.*PI)
    3 CONTINUE
   30 IF(ISORCE.EQ.1.OR.ISORCE.EQ.4) GO TO 40
      IF(ISORCE.EQ.2.OR.ISORCE.EQ.5) GO TO 50
      IF(ISORCE.EQ.3.OR.ISORCE.EQ.6) GO TO 60
C     SPATIALLY UNIFORM SOURCE
   40 DO 41 KX=1,NX
      DO 41 L=1,LMAX
      FSUBL(L,KX)=SL(L)
   41 CONTINUE
      RETURN
C     SPATIAL DELTA FUNCTION (3 CELLS)
   50 DO 51 L=1,LMAX
      FSUBL(L,KXSRCE)=0.5*SL(L)
      FSUBL(L,KXSRCE+1)=0.25*SL(L)
      FSUBL(L,KXSRCE-1)=0.25*SL(L)
   51 CONTINUE
      RETURN
C     GAUSSIAN SPATIAL DISTRIBUTION
   60 XX=-DELX
      SPI=SQRT(PI)
      BSIG=ASIG/SPI
      DO 61 KX=1,NX
      XX=XX+DELX
      AB=ASIG*(XX-XSOURCE)
      VALUE=BSIG*EXP(-AB**2)
      DO 61 L=1,LMAX
      FSUBL(L,KX)=VALUE*SL(L)
   61 CONTINUE
      RETURN
      ENTRY SOURCE
      IF(ISTART.NE.0)GO TO 110
      REWIND 5
      DO 111 L=1,LMAX
      DO 111 KX=1,NX
      SLC(L,KX)=0.0
  111 CONTINUE
  110 DO 103 L=1,LMAX
      DO 103 KX=1,NX
      SDRV(L,KX)=SLC(L,KX)
  103 CONTINUE
      DO 101 L=1,LMAX
      READ (5) (SLC(L,KX),KX=1,NX)
      FSUBL(L,1)=(SLC(L,2)-SLC(L,1))/DELX
      NX1=NX-1
      FSUBL(L,NX)=(SLC(L,NX)-SLC(L,NX1))/DELX
      DO 102 KX=2,NX1
      FSUBL(L,KX)=0.5*(SLC(L,KX+1)-SLC(L,KX-1))/DELX
  102 CONTINUE
  101 CONTINUE
      IF(ISTART.GT.0)GO TO 104
      DO 106 L=1,LMAX
      READ(5)(SDRV(L,KX),KX=1,NX)
      DO 107 KX=1,NX
      SDRV(L,KX)=(SDRV(L,KX)-SLC(L,KX))/DS
  107 CONTINUE
  106 CONTINUE
      ISTART=1
      DO 116 L=1,LMAX
      BACKSPACE 5
  116 CONTINUE
      GO TO 108
  104 DO 105 L=1,LMAX
      DO 105 KX=1,NX
      SDRV(L,KX)=(SLC(L,KX)-SDRV(L,KX))/DS
  105 CONTINUE
  108 RETURN
      END
```

Subroutine LEGEXP-Flux angular distribution

```
      SUBROUTINE LEGEXP
      COMMON PL(201,51),AMU(201),FSUBL(51,101),FOLD(51,101),PGAUS(6,51),
     XANGMU(6),Q(51),Z(51),SL(51),FOCHRG(101),BACHRG(101),CHRG(101),
     /EDPOS(101),LMAX,NX,LAMBDA,ETA,ISCAT,DS,JSMAX,JSKIP,MUZERO,ISORCE,
     ZKXSRCE,IDBG,IPRNT,METAL,WF(101),WB(101),WNET(101),DELX,ASIG,
     WXSOURCE,EZERO,ECUT,ENRGY,SDRV(51,101),SLC(51,101)
      REAL MUZERO,LAMBDA
      DIMENSION Y(51),F(201)
      DO 10 I=1,6
      SUM=0.
      DO 11 KX=1,NX
      F(KX)=0.
      DO 12 L=1,LMAX
      AL=0.5*FLOAT(2*L-1)
      F(KX)=F(KX)+AL*PGAUS(I,L)*FSUBL(L,KX)
   12 CONTINUE
      SUM=SUM+F(KX)
   11 CONTINUE
      PRINT 13,ANGMU(I)
   13 FORMAT(/1X,*ANGULAR FLUX, MU=*,E13.6)
      IF(IDBG.NE.0) PRINT 14,(F(KX),KX=1,NX)
   14 FORMAT(1X,10E13.5)
      PRINT 15,SUM
   15 FORMAT(1X,*ANGULAR FLUX SUMMED OVER X=*,E15.9)
   10 CONTINUE
      RETURN
      ENTRY POLANG
      ANGMU(1)=.2386192
      ANGMU(3)=.6612094
      ANGMU(5)=.9324695
      ANGMU(2)=-ANGMU(1)
      ANGMU(4)=-ANGMU(3)
      ANGMU(6)=-ANGMU(5)
      DO 1 I=1,6
      X=ANGMU(I)
      CALL LEP(Y,X,50)
      DO 2 L=1,51
      PGAUS(I,L)=Y(L)
    2 CONTINUE
    1 CONTINUE
      RETURN
      END
```

Subroutine TPLOT-Plots computed results

```
      SUBROUTINE TPLOT(NX,M,J,KALCOMP)
      DIMENSION X(201),Y(201),VLAB1(19),VLAB2(25),HLABL(16),
     1VLAB(30)
      DIMENSION NVCHAR(4),MCHR(5)
      DIMENSION TITLX(3),TITLY(3),TITL1(3),TITL2(3),PROGID(3)
      DATA MCHR/1H*,1HX,1H+,1HO,1H./
      DATA HLABL/1HP,1HO,1HS,1HI,1HT,1HI,1HO,1HN,1H ,1HI,1HN,1H ,1HS,
     X1HL,1HA,1HB/
      DATA VLAB2/1HE,1HN,1HE,1HR,1HG,1HY,1H ,1HD,1HE,1HP,1HO,1HS,1HI,1HT
     X,1HI,1HO,1HN,1H ,1HP,1HR,1HO,1HF,1HI,1HL,1HE/
      DATA VLAB1/1HC,1HH,1HA,1HR,1HG,1HE,1H ,1HD,1HI,1HS,1HT,1HR,1HI,1HB
     X,1HU,1HT,1HI,1HO,1HN/
      DATA NHCHAR,NVCHAR,IMAGE,NCHAX,NVCHX/16,19,25,19,25,0,30,28/
      DATA XSCALE,YSCALE,NPLOT,NPRNT/6.0,8.0,0,0/
      DATA TITLX/10HPOSITION I,10HN SLAB (RA,10HNGE UNITS)/
      DATA TITL1/10HCHARGE DIS,10HTRIBUTION ,8HFUNCTION/
      DATA TITL2/10HENERGY DEP,10HOSITION PR,8HOFILE   /
      ICHR=0
      YMAX=0.0
      READ(J) JS,(X(KX),KX=1,NX)
   20 READ(J) JS,(Y(KX),KX=1,NX)
      IF(EOF(J))21,22
   22 DO 13 KX=1,NX
      IF(Y(KX).GT.YMAX)YMAX=Y(KX)
   13 CONTINUE
      GO TO 20
   21 REWIND J
      WIDTH=X(NX)
      IF(KALCOMP.EQ.0) GO TO 15
      IF(NPLOT.EQ.0) READ 17,(PROGID(I),I=1,3)
   17 FORMAT(3A10)
      KK=IFIX(ALOG10(YMAX))
      IF(KK.GE.0)KK=KK+1
      CYTOP=FLOAT(KK)
      CYBOT=CYTOP-5.0
      NVCH=NVCHAR(J)
      DO 16 N=1,3
      IF(J.EQ.1.OR.J.EQ.3) TITLY(N)=TITL1(N)
      IF(J.EQ.2.OR.J.EQ.4) TITLY(N)=TITL2(N)
   16 CONTINUE
      NPLOT=NPLOT+1
      XBT=0.0
      CALL SMLGPLT(X,Y,NX,WIDTH,XBT,CYTOP,CYBOT,XSCALE,YSCALE,TITLX,
     XNCHAX,TITLY,NVCHX,PROGID,NPRNT,NPLOT)
   15 YMIN=0.0
      CALL PLOT2(IMAGE,WIDTH,0.0,YMAX,YMIN,IC)
    1 READ(J)JS,(X(KX),KX=1,NX)
   10 READ(J)JS,(Y(KX),KX=1,NX)
      IFLG=0
      IF(EOF(J).NE.0)IFLG=1
      IF(KALCOMP.EQ.0)GO TO 18
      IF(IFLG.EQ.1) GO TO 19
      CALL CURVE  (X,Y,NX,WIDTH,XBT,CYTOP,CYBOT,XSCALE,YSCALE,TITLX.
     XNCHAX,TITLY,NVCHX,PROGID,NPRNT,NPLOT)
   18 ICHR=ICHR+1
      IF(ICHR.GT.5)ICHR=1
      MARK=MCHR(ICHR)
      IF(IFLG.EQ.1)GO TO 3
   12 CALL PLOT3(MARK,X,Y,NX,IC)
      GO TO 10
    3 WRITE(M,201)
  201 FORMAT(*1*)
      NVCH=NVCHAR(J)
      DO 14 N=1,NVCH
      IF(J.EQ.1.OR.J.EQ.3)VLAB(N)=VLAB1(N)
      IF(J.EQ.2.OR.J.EQ.4)VLAB(N)=VLAB2(N)
   14 CONTINUE
      CALL PLOT4(NVCH,VLAB,NHCHAR,HLABL,IC)
      RETURN
      END
```

SubroutinesPLOT2, PLOT3, PLOT4 are too long to include here. They are part of a printer plot package written by Mr. Michael Clark, Prof. Brice Carnahan and Mr. Michael Duncan of the University of Michigan. These may be obtained from Prof. Carnahan, Chemical Eng. Dept., U. Mich., Ann Arbor, Mich. 48104 (phone 313-764-3366) or from S. Woolf, Arcon Corp., Waltham, Mass. (phone 890-3330)

Subroutine SMLGPLT-Semi-log plotting program

```
      SUBROUTINE SMLGPLT(X,Y,NXY,XTOP,XBOT,CYTOP,CYBOT,XSCALE,YSCALE,
     1TITLX,NCHARX,TITLY,NCHARY,PROGID,NPRNT,NPLOT)
      DIMENSION X(1),Y(1),TITLX(3),TITLY(3),PROGID(3),XPLOT(201),YPLOT(2
     101),YP(201)
      DATA XMAX,YMAX,FACTOR/150.,11.,1./
      IF(NPLOT.EQ.1)CALL PLTID3(PROGID,XMAX,YMAX,FACTOR)
      IF(NPLOT.EQ.1)CALL PLOT(4.,1.,-3)
      KRV=0
      XMOVE=XSCALE+4.
      IF(NPLOT.NE.1)CALL PLOT(XMOVE,0.,-3)
      DX=(XTOP-XBOT)/XSCALE
      DY=(CYTOP-CYBOT)/YSCALE
      ICALL =1
      CALL AXUS(XBOT,XTOP,1,DX,XSCALE,YSCALE,TITLX,NCHARX,ICALL)
      ICALL=2
      CALL AXUS(CYBOT,CYTOP,2,DY,XSCALE,YSCALE,TITLY,NCHARY,ICALL)
      RETURN
C
C
      ENTRY CURVE
C
C
      KRV=KRV+1
      YHI=(10.)**CYTOP
      YLO=(10.)**CYBOT
      KL=0
      DO 1 N=1,NXY
      YY=Y(N)
      XX=X(N)
      IF(YY.GT.YHI) GO TO 1
      IF(YY.LT.YLO) GO TO 1
      IF(XX.GT.XTOP) GO TO 1
      IF(XX.LT.XBOT) GO TO 1
      KL=KL+1
      YP(KL)=YY
      XPLOT(KL)=XX
      YPLOT(KL)=ALOG10(YY)-CYBOT
    1 CONTINUE
      NPT=KL
      PRINT 10,NPLOT,KRV
   10 FORMAT(//1X,*PLOT NO.*,I3,2X,*CURVE NO.*,I3)
      IF(NPRNT.EQ.0) GO TO 2
      PRINT 11,(XPLOT(KL),YP(KL),KL=1,NPT)
   11 FORMAT(1X,10E11.5)
    2 CALL LINE(XPLOT,YPLOT,NPT,1,0,0,0.,DX,0.,DY,0.08)
      RETURN
      END
```

Subroutine AXUS-draws logarithmic axis
(This subroutine has been superceded by an AFGL supplied routine called AXLOG which may be accessed from the AFGL plot utility library)

```
      SUBROUTINE AXUS(X1,X2,LXY,D,XSCALE,YSCALE,TEXT,NT,ICALL)
      DIMENSION TEXT(3),E(14),NE(14),YC(14),ALG(8)
      DATA TEX1/3HX10/
      DATA TEX2/4HX10-/
      DATA E/2H10,1H9,1H8,1H7,1H6,1H5,1H4,1H3,1H2,1H1,1H0,1H1,1H2,1H3/
      DATA NE/2,13*1/
      DATA YC/10.,9.,8.,7.,6.,5.,4.,3.,2.,1.,0.,1.,2.,3./
      M=3
      IF(ICALL.EQ.1)M=-3
      GO TO (100,200),LXY
C      X AXIS
  100 XLW=X1/D
      XH=XSCALE+XLW
      CALL PLOT(0.,0.,4)
      CALL PLOT(XLW,0.,3)
      CALL PLOT(XH,0.,2)
      CALL PLOT(XLW,0.,3)
      EPS=1.E-6
      XDIF=ABS((X2-X1)*(1.+EPS))
      IL=IFIX(ALOG10(XDIF))
      IF(IL.LT.0)IL=IL-1
      FCTR=10.0**IL
      XTLAB=X2/FCTR
      XBLAB=X1/FCTR
      ITEST=IFIX(XTLAB-XBLAB+EPS)
      DD=0.1*FLOAT(ITEST)
      D1=(XTLAB-XBLAB)/XSCALE
      DDD=DD/D1
      LMAX=1+IFIX((XTLAB-XBLAB+EPS)/DD)
      XL=XLW-DDD
      XLL=XBLAB-DD
      DO 102 L=1,LMAX
      XL=XL+DDD
      XLL=XLL+DD
      CALL SYMBOL(XL,0.,0.1,13,0.0,-1)
      XL1=XL-.15
      CALL NUMBER(XL1,-.15,0.08,XLL,0.,1)
  102 CONTINUE
      CALL PLOT(0.95,-0.4,3)
      XAT=FLOAT(NT)*0.15
      XSTART=XLW+0.5*(XH-XLW-XAT)
      CALL SYMBOL(XSTART,-0.5,0.15,TEXT,0.,NT)
      FA=FLOAT(IABS(IL))
      DO 306 II=1,11
      IF(ABS(FA-YC(II)).LE.EPS)GO TO 307
  306 CONTINUE
  307 S=E(II)
      J=NE(II)
      XAT=0.15*5.
      XSTART=XLW+0.5*(XH-XLW-XAT)
      IF(IL.GE.0)GO TO 308
      TEX3=TEX2
      NEX=4
      GO TO 309
  308 TEX3=TEX1
      NEX=3
  309 CALL SYMBOL(XSTART,-.75,.15,TEX3,0.,NEX)
```

```
      XSTART=XSTART+0.15*FLOAT(NEX)
      CALL SYMBOL(XSTART,-.65,.1,S,0.,J)
      RETURN
C     Y AXIS
  200 CALL PLOT(XLW,0.,M)
      CALL PLOT(XLW,YSCALE,2)
      CALL PLOT(0.,0.,3)
      DO 50 I=1,8
      ALG(I)=ALOG10(FLOAT(I)+1.0)
   50 CONTINUE
      LY=IFIX(X2-X1+0.0001)
      DO 201 L=1,LY
      LL=L-1
      YL=FLOAT(LL)/D
      XL7=XLW-0.05
      CALL SYMBOL(XL7,YL,0.2,13,90.,-1)
      YL1=FLOAT(L)/D
      DY=YL1-YL
      DO 204 K=1,8
      YL2=YL+ALG(K)*DY
      CALL SYMBOL(XLW,YL2,0.1,13,90.,-1)
  204 CONTINUE
  201 CONTINUE
      CALL PLOT(XH,0.,3)
      XP=XH-0.05
      XP1=XH-0.025
      DO 205 L=1,LY
      LL=L-1
      YL=FLOAT(LL)/D
      CALL SYMBOL(XP,YL,0.2,13,90.,-1)
      YL1=FLOAT(L)/D
      DY=YL1-YL
      DO 206 K=1,8
      YL2=YL+ALG(K)*DY
      CALL SYMBOL(XP1,YL2,0.1,13,90.,-1)
  206 CONTINUE
  205 CONTINUE
      XL=XLW-0.41
      XL1=XLW-0.25
      XL2=XLW-0.18
      XL0=XLW-.33
      LY=LY+1
      CY=ABS(X1)
      DO 207 I=1,11
      IF(ABS(CY-YC(I)).LE.EPS)IC1=I
  207 CONTINUE
      IC=IC1-1
      DO 202 L=1,LY
      LL=L-1
      YL=FLOAT(LL)/D-.05
      YL1=YL+0.1
      CALL SYMBOL(XL,YL,0.08,49,0.0,-1)
      CALL SYMBOL(XL0,YL,0.08,48,0.0,-1)
      IC=IC+1
      IF(IC.GE.11) GO TO 203
      CALL SYMBOL(XL1,YL1,0.07,45,0.0,-1)
  203 S=E(IC)
```

```
      J=NE(IC)
      CALL SYMBOL(XL2,YL1,0.08,S,0.,J)
  202 CONTINUE
      YAT=FLOAT(NT)*0.15
      YSTART=0.5*(YSCALE-YAT)
      CALL PLOT(-0.6,2.0,3)
      XLL=XLW-0.45
      CALL SYMBOL(XLL,YSTART,0.15,TEXT,90.,NT)
      RETURN
      END
```

APPENDIX 2

Programs GAUSQN
ANISO
GNLEG
LOOK

Program GAUSQN

Program GAUSQN computes the reduced source functions, $G_n(q)$ for isotropically scattered particles emanating from a forward-directed point source in an infinite medium.

The routines are

1) GAUSQN - Main program, performs Gaussian quadrature integration of isotropic G_n recursion relation.

2) Subroutine SETUP - Reads in Gauss quadrature weights and ordinates.

3) Subroutine SPCOEF - Computes spline interpolation coefficients.

4) Function SPLINE - Evaluates G_n functions at Gauss quadrature ordinates.

5) Functions AJ, AI - Evaluates portions of integrands in recursion integrals.

6) Subroutine LINPLT - Plots results on linear scale.

7) Subroutine AXIL - Draws and labels linear axes.

Program GAUSQN-Gaussian quadrature integration of G_n recursion relation: Isotropic scattering

```
      PROGRAM GAUSQN(INPUT,OUTPUT,TAPE1,TAPE2)
C
C
C   THIS PROGRAM CALCULATES THE N-TH SCATTERED SOURCE FUNCTIONS GN(Q) AS
C   A FUNCTION OF Q(=X/S) FOR ISOTROPIC SCATTERING.  IT BEGINS WITH
C   G2(Q) GIVEN AS AN ANALYTICAL FUNCTION OF Q.  G3(Q) IS THEN COMPUTED
C   FROM G2 IN ACCORDANCE WITH THE INTEGRAL EQUATION FOR GN+1(Q) IN
C   TERMS OF GN(Q) GIVEN BY GARTH IN HIS NOTES OF 5 SEPT.1977 PAGE P2.
C   THE IN AND JN FUNCTIONS ARE COMPUTED HERE IN FUNCTION SUBPROGRAMS AI
C   AND AJ RESPECTIVELY.  THE NUMERICAL INTEGRATION IS PERFORMED USING A
C   32 POINT GAUSSIAN QUADRATURE.  THE ARRAY G3(Q) IS STORED TEMPORARILY
C   IN CORE AND IS ALSO WRITTEN OUT TO A MASS STORAGE FILE FOR
C   SUBSEQUENT PLOTTING.  THE COMPUTED ARRAY G3(Q) IS THEN USED TO
C   OBTAIN G4(Q), WHICH IN TURN IS USED TO OBTAIN G5(Q), AND SO ON.
C   ALL NUMERICAL INTEGRATIONS ARE DONE USING THE 32 POINT GAUSSIAN
C   QUADRATURE.  HOWEVER, SINCE THE GN FUNCTIONS EXIST ONLY IN TABULAR
C   FORM FOR N.GT.2, THE VALUES OF THESE FUNCTIONS AT THE GAUSSIAN
C   ORDINATES ARE OBTAINED BY SPLINE INTERPOLATION (SHAMPINE AND ALLEN
C   NUMERICAL COMPUTING, SAUNDERS, 238-239(1973)) ON THE TABLES. EACH
C   GN(Q) ARRAY (201 VALUES OF Q EQUISPACED FROM -1 TO +1) IS STORED ON
C   MASS STORAGE FOR SUBSEQUENT PLOTTING.
C
C    THE PROGRAM CONTAINS AN OPTION TO RENORMALIZE THE COMPUTED GN
C   ARRAYS AFTER EACH ARRAY IS OBTAINED.  THIS IS DONE TO REDUCE ERROR
C   PROPAGATION INTRODUCED BY NUMERICAL INTEGRATION INACCURACIES.
C
C    THERE ARE ALSO OPTIONS TO PUNCH THE GN ARRAYS ON CARDS AND TO PLOT
C   THE GN CURVES ON THE CALCOMP.
C
      DIMENSION Q(201),W(32),A(32),X(32),GN(201),S(201),INDEX(201)
      DIMENSION TITLX(3),TITLY(3),PROGID(3)
      DATA TITLY/10HG(Q) ISOTR,10HOPIC SCATT,5HERING/
      DATA TITLX/1HQ/
C
C
C   INPUT PARAMETERS:
C       NMAX = HIGHEST ORDER OF SCATTERING (N) TO BE COMPUTED
C       INORM = RE-NORMALIZATION OPTION PARAMETER (0=NO RENORMALIZATION,
C               1 = RE-NORMALIZATION)
C       IPLOT = PLOT OPTION PARAMETER (0=NO PLOTS, 1=PLOTS)
C       IPUNCH = PUNCH OPTION PARAMETER (0=NO PUNCH, 1=PUNCH)
C
      READ 1,NMAX,INORM,IPLOT,IPUNCH
    1 FORMAT(4I5)
C
C   SUBROUTINE SETUP READS IN THE 32 POINT GAUSSIAN QUADRATURE WEUGHTS,A
C   , AND ORDINATES,W.  IT ALSO COMPUTES THE Q ARRAY.
C
      CALL SETUP(W,A,X,Q)
      SUMT=0.
      DQ=.01
      MAX=201
      N=3
      M=N-1
C
C   COMPUTE G3(Q) FOR 201 VALUES OF Q FROM G2(Q).
C
```

```
      DO 100 I=1,201
      E1=0.
      IF(I.EQ.201)GO TO 31
C
C     CONVERT GAUSSIAN ORDINATES, W(J), TO THE VARIABLE OF INTEGRATION FOR
C     THE FIRST INTEGRAL IN GARTH'S EXPRESSION FOR GN+1(Q) (5SEPT. 1977,
C     PAGE R2).
C
C     THE SECOND INTEGRAL IN GARTH'S EXPRESSION FOR GN+1(Q) (5SEPT. 1977,
C     CONVERT GAUSSIAN ORDINATES, W(J), TO THE VARIABLE OF INTEGRATION FOR
C     PAGE R2).
C
      DO 101 J=1,32
  101 X(J)=0.5*((1.+Q(I))+(1.-Q(I))*W(J))
C
C     COMPUTE THE VALUE OF THE FIRST INTEGRAL (DENOTED AS E1 IN CODE).
C
      DO 102 J=1,32
      E1=E1+0.5*(1.-Q(I))*A(J)*ALOG(2./(1.-X(J)))*AI(M,Q(I),X(J),I,J)
  102 CONTINUE
C
C     COMPUTE THE VALUE OF THE SECOND INTEGRAL (DENOTED AS E2 IN CODE).
C
   31 E2=0.
      IF(I.EQ.1) GO TO 32
      DO 103 J=1,32
  103 X(J)=0.5*((Q(I)-1.)+(1.+Q(I))*W(J))
      DO 104 J=1,32
      E2=E2+0.5*(1.+Q(I))*A(J)*ALOG(2./(1.-X(J)))*AJ(M,Q(I),X(J),I,J)
  104 CONTINUE
   32 GN(I)=0.5*(E1+E2)
      SUMT=SUMT+GN(I)*DQ
C
C     PRINT G3 ARRAY
C
      PRINT 305,I,Q(I),GN(I),SUMT,E1,E2
  305 FORMAT(1X,I5,F5.2,4(5X,F10.5))
  100 CONTINUE
      IF(INORM.EQ.0)GO TO 120
      PRINT 209,N
C
C     RE-NORMALIZATION LOOP FOR G3
C
      DO 110 I=1,201
  110 GN(I)=GN(I)/SUMT
C
C     PRINT RE-NORMALIZED G3 ARRAY
C
      PRINT 306,(I,Q(I),GN(I),I=1,201)
C
C     WRITE G3 ARRAY ON MASS STORAGE
C
  120 WRITE(1)N,(GN(I),I=1,201)
C
C     LOOP TO CALCULATE THE GN(Q) ARRAYS FOR N.GE.3
C
      DO 50 N=4,NMAX
      M=N-1
      AN=0.5*FLOAT(N-1)
      SUMT=0.
      PRINT 77,N
   77 FORMAT(*1*,5X,*N=*,I2)
C
C   GET SPLINE COEFFICIENTS OF PREVIOUSLY CALCULATED GN-1(Q).
C
      CALL SPCOEF(MAX,Q,GN,S,INDEX)
C
C   COMPUTE THE VALUE OF THE FIRST INTEGRAL (DENOTED AS E1 IN CODE).
C
      DO 200 I=1,201
      E1=0.
      IF(I.EQ.201)GO TO 41
C
C   CONVERT GAUSSIAN ORDINATES, W(J), TO THE VARIABLE OF INTEGRATION FOR
C   THE FIRST INTEGRAL IN GARTH*S EXPRESSION FOR GN+1(Q) (5SEPT. 1977,
C   PAGE R2).
C
      DO 201 J=1,32
  201 X(J)=0.5*((1.+Q(I))+(1.-Q(I))*W(J))
      DO 202 J=1,32
      XX=X(J)
C
C   DO SPLINE INTERPOLATION TO OBTAIN GN VALUES AT GAUSSIAN ORDINATES.
C
      G=SPLINE(MAX,Q,GN,S,INDEX,XX)
      E1=E1+0.5*(1.-Q(I))*A(J)*G*AI(M,Q(I),X(J),I,J)
  202 CONTINUE
C
C   COMPUTE THE VALUE OF THE SECOND INTEGRAL (DENOTED AS E2 IN CODE).
C
   41 E2=0.
      IF(I.EQ.1) GO TO 42
C
C   CONVERT GAUSSIAN ORDINATES, W(J), TO THE VARIABLE OF INTEGRATION FOR
C   THE SECOND INTEGRAL IN GARTH*S EXPRESSION FOR GN+1(Q) (5SEPT. 1977,
C   PAGE R2).
C
      DO 203 J=1,32
  203 X(J)=0.5*((Q(I)-1.)+(1.+Q(I))*W(J))
      DO 204 J=1,32
      XX=X(J)
C
C   DO SPLINE INTERPOLATION TO OBTAIN GN VALUES AT GAUSSIAN ORDINATES.
C
      G=SPLINE(MAX,Q,GN,S,INDEX,XX)
      E2=E2+0.5*(1.+Q(I))*A(J)*G*AJ(M,Q(I),X(J),I,J)
  204 CONTINUE
   42 GN(I)=AN*(E1+E2)
      SUMT=SUMT+GN(I)*DQ
C
C   PRINT GN ARRAY
C
      PRINT 305,I,Q(I),GN(I),SUMT,E1,E2
  200 CONTINUE
```

```
C
C     RE-NORMALIZATION LOOP FOR GN
C
      IF(INORM.EQ.0) GO TO 220
      PRINT 209,N
      DO 210 I=1,201
  210 GN(I)=GN(I)/SUMT
C
C     PRINT RE-NORMALIZED GN ARRAY
C
      PRINT 306,(I,Q(I),GN(I),I=1,201)
  306 FORMAT(6(I5,F5.2,2X,F10.5))
  209 FORMAT(*1*,5X,*N=*,I3,5X,*RENORMALIZED*)
C
C     WRITE GN ARRAY ON MASS STORAGE
C
  220 WRITE(1)N,(GN(I),I=1,201)
   50 CONTINUE
C
C     PLOT AND PUNCH CODE SECTION
C
      NPLOT=1
      REWIND 1
      IF(IPLOT.EQ.0)GO TO 61
      READ 17,(PROGID(I),I=1,3)
   17 FORMAT(3A10)
      CALL LINPLT(Q,GN,201,1.0,-1.0,2.0,0.0,6.0,8.0,TITLX,25,TITLY,1,
     1PROGID,0,NPLOT)
      DO 60 N=3,NMAX
      READ(1)NN,(GN(I),I=1,201)
      CALL LCURVE(Q,GN,201,1.0,-1.0,2.0,0.0,6.0,8.0,TITLX,25,TITLY,1,
     1PROGID,0,NPLOT)
   60 CONTINUE
      CALL ENDPLT
   61 IF(IPUNCH.EQ.0)GO TO 71
      IF(INORM.EQ.0)WRITE(2,62)
      IF(INORM.NE.0)WRITE(2,63)
   62 FORMAT(1X,*GN(N=2,10)  NOT RENORMALIZED*)
   63 FORMAT(1X,*GN(N=2,10)  RENORMALIZED*)
      REWIND 1
      DO 64 N=3,NMAX
      WRITE(2,66)N
   66 FORMAT(1X,*N=*,I10)
      READ(1)NN,(GN(I),I=1,201)
      WRITE(2,67)(GN(I),I=1,201)
   67 FORMAT(10F8.6)
   64 CONTINUE
   71 STOP
      END
```

Subroutine SETUP-Read Gauss quadrature ordinates and weights

```
      SUBROUTINE SETUP(W,A,X,Q)
      DIMENSION W(32),A(32),X(32),Q(201)
C
C     READ GAUSSIAN ORDINATES,W.
C
      READ 1,(W(I),I=1,16)
C
C     READ GAUSSIAN WEIGHTS,A.
C
      READ 1,(A(I),I=1,16)
    1 FORMAT(4F20.0)
C
C     PLACE GAUSSIAN ORDINATES IN PROPER ORDER.
C
      DO 2 I=1,16
      X(I)=W(17-I)
    2 X(I+16)=-W(I)
      DO 3 I=1,32
    3 W(I)=X(I)
C
C     PLACE GAUSSIAN WEIGHTS IN PROPER ORDER.
C
      DO 4 I=1,16
      X(I)=A(17-I)
    4 X(I+16)=A(I)
      DO 5 I=1,32
    5 A(I)=X(I)
      Q(1)=-1.
      DQ=.01
C
C     COMPUTE THE Q (=X/S) ARRAY.
C
      DO 6 I=2,201
    6 Q(I)=Q(I-1)+DQ
      RETURN
      END
```

INPUT DATA:

```
.0483076656877383162.1444719615827964935.2392873622521370745.3318686022821276498
.4213512761306353454.5068999089322293900.5877157572407623290.6630442669302152009
.7321821187402896804.7944837959679424069.8493676137325699701.8963211557660521239
.9349080759377396892.9647622555875064308.9856115115452683354.9972638618494815635
.0965400885147278005.0956387200792748594.0938443990808045656.0911738786957638847
.0876520930044038111.0833119242269467552.0781938957870703064.0723457941088485062
.0658222227763618468.0586840934785355471.0509980592623761762.0428358980222266806
.0342738629130214331.0253920653092620595.0162743947309056706.0070186100094700966
```

Functions AJ, AI-evaluates J_n and I_n functions

```
      FUNCTION AJ(N,Q,R,I,J)
C
C     THIS PROGRAM COMPUTES THE JN FUNCTION (GARTH NOTES,5 SEPT.1977,P.R2)
C
      DIMENSION Y(201,32)
      DATA Y/6432*0./
      NN=N-1
      ALPHA=ALOG((1.-R)/(Q-R))
      X=(1.-Q)/(1.-R)
    1 BJ=FLOAT(NN)
      Y(I,J)=Y(I,J)+(X**NN)/BJ
      AJ=ALPHA-Y(I,J)
      RETURN
      END
```

```
      FUNCTION AI(N,Q,R,I,J)
C
C     THIS PROGRAM COMPUTES THE IN FUNCTION (GARTH NOTES,5 SEPT.1977,P.R2)
C
      DIMENSION Y(201,32)
      DATA Y/6432*0./
      NN=N-1
      ALPHA=ALOG((1.+R)/(R-Q))
      X=(1.+Q)/(1.+R)
    1 AJ=FLOAT(NN)
      Y(I,J)=Y(I,J)+(X**NN)/AJ
      AI=ALPHA-Y(I,J)
      RETURN
      END
```

Subroutine SPCOEF-Evaluate spline coefficients-

Algorithm and code written by L. F. Shampine and R. C. Allen, <u>Numerical Computing</u>, Saunders, Pp. 54-57 (1973)

```
      SUBROUTINE SPCOEF(I,J,S2,S3,KXY,X,Y,F,IMAX,JMAX)
C
C     THIS SUBROUTINE EVALUATES THE SPLINE COEFFICIENTS FOR A FIVE
C     POINT CUBIC SPLINE USING THE SHAMPINE RECURSIVE ALGORITHM (OP.CIT.).
C     IT RETURNS THE SPLINE COEFFICIENTS S2 AND S3 CORRESPONDING TO THE
C     THIRD AND FOURTH POINTS.  ( THE COEFFICIENTS S0 AND S4 CORRESPONDING
C     TO THE FIRST AND FIFTH POINTS ARE ZERO, AND S1 AT THE SECOND POINT
C     IS NOT NEEDED.)
C
      DIMENSION X(IMAX),Y(JMAX),F(IMAX,JMAX)
      DIMENSION Z(5),FF(5)
      IF(KXY.EQ.1)1,2
    1 DO 10 K=1,5
      KK=I+K-1
      Z(K)=X(KK)
   10 FF(K)=F(KK,J)
      GO TO 3
    2 DO 20 K=1,5
      KK=J+K-1
      Z(K)=Y(KK)
   20 FF(K)=F(I,KK)
    3 H0=Z(2)-Z(1)
      H1=Z(3)-Z(2)
      H2=Z(4)-Z(3)
      H3=Z(5)-Z(4)
      D1=6./H1*((FF(3)-FF(2))/H1-(FF(2)-FF(1))/H0)
      D2=6./H2*((FF(4)-FF(3))/H2-(FF(3)-FF(2))/H1)
      D3=6./H3*((FF(5)-FF(4))/H3-(FF(4)-FF(3))/H2)
      RHO2=-.5/(1.+H0/H1)
      RHO3=-1./(RHO2*H1/H2+2.*(1.+H1/H2))
      RHO4=-1./(RHO3*H2/H3+2.*(1.+H2/H3))
      TAU2=-D1*RHO2
      TAU3=-(D2-H1/H2*TAU2)*RHO3
      S3=-(D3-H2/H3*TAU3)*RHO4
      S2=RHO3*S3+TAU3
      RETURN
      END
```

Function SPLINE- performs spline interpolation-
Algorithm and code written by L. F. Shampine and R. C. Allen, Numerical Computing , Saunders, Pp. 54-57 (1973)

```
      FUNCTION SPLINE(N,XN,FN,S,INDEX,X)
      DIMENSION XN(N),FN(N),S(N),INDEX(N)
      I1=INDEX(1)
      IF(X.GE.XN(I1))GO TO 1
      I2=INDEX(2)
      H1=XN(I2)-XN(I1)
      SPLINE=FN(I1)+(X-XN(I1))*((FN(I2)-FN(I1))/H1-H1*S(2)/6.0)
      RETURN
    1 IN=INDEX(N)
      IF(X.LE.XN(IN))GO TO 2
      INM1=INDEX(N-1)
      HNM1=XN(IN)-XN(INM1)
      SPLINE=FN(IN)+(X-XN(IN))*((FN(IN)-FN(INM1))/HNM1+HNM1*S(N-1)/6.0)
      RETURN
    2 DO 3 I=2,N
      II=INDEX(I)
      IF(X.LE.XN(II))GO TO 4
    3 CONTINUE
    4 L=I-1
      IL=INDEX(L)
      ILP1=INDEX(L+1)
      A=XN(ILP1)-X
      B=X-XN(IL)
      HL=XN(ILP1)-XN(IL)
      SPLINE=A*S(L)*(A**2/HL-HL)/6.0+B*S(L+1)*(B**2/HL-HL)/6.0
     1      +(A*FN(IL)+B*FN(ILP1))/HL
      RETURN
      END
```

Subroutine LINPLT- Plots results on linear scale

```
      SUBROUTINE LINPLT(X,Y,NXY,XTOP,XBOT,YTOP,YBOT,XSCALE,YSCALE,TITLX,
     XNCHARX,TITLY,NCHARY,PROGID,NPRNT,NPLOT)
      DIMENSION X(1),Y(1),TITLX(3),TITLY(3),PROGID(3),XPLOT(201),YPLOT(2
     101)
      DATA XMAX,YMAX,FACTOR/150.,11.,1./
      IF(NPLOT.EQ.1)CALL  PLTID3(PROGID,XMAX,YMAX,FACTOR)
      IF(NPLOT.EQ.1)CALL  PLOT(4.,1.,-3)
      KRV=0
      XMOVE=XSCALE+4.
      IF(NPLOT.NE.1)CALL  PLOT(XMOVE,0.,-3)
      DX=(XTOP-XBOT)/XSCALE
      DY=(YTOP-YBOT)/YSCALE
      ICALL=1
      CALL  AXIL(XBOT,XTOP,1,DX,XSCALE,YSCALE,TITLX,NCHARX,ICALL)
      ICALL=2
      CALL  AXIL(YBOT,YTOP,2,DY,XSCALE,YSCALE,TITLY,NCHARY,ICALL)
      RETURN
C
C
      ENTRY  LCURVE
C
C
      KRV=KRV+1
      KL=0
      DO 1 N=1,NXY
      YY=Y(N)
      XX=X(N)
      IF(YY.GT.YTOP)GO TO 1
      IF(YY.LT.YBOT)GO TO 1
      IF(XX.GT.XTOP) GO TO 1
      IF(XX.LT.XBOT) GO TO 1
      KL=KL+1
      XPLOT(KL)=XX
      YPLOT(KL)=YY
    1 CONTINUE
      NPT=KL
      IF(NPT.EQ.0)RETURN
      PRINT 10,NPLOT,KRV
   10 FORMAT(//1X,*PLOT NO.*,I3,2X,*CURVE NO.*,I3)
      IF(NPRNT.EQ.0) GO TO 2
      PRINT 11,(XPLOT(KL),YPLOT(KL),KL=1,NPT)
   11 FORMAT(1X,10E11.5)
    2 CALL  LINE(XPLOT,YPLOT,NPT,1,0,0,0.,DX,0.,DY,0.08)
      RETURN
      END
```

Subroutine AXIL- Draws and labels linear axes (this routine has been superceded by an AFGL supplied subroutine called AXISL which exists in the AFGL plotting utility package.)

```
      SUBROUTINE AXIL(X1,X2,LXY,D,XSCALE,YSCALE,TEXT,NT,ICALL)
      DIMENSION TEXT(3),E(41),NE(41),YC(41)
      DATA TEX1/3HX10/
      DATA TEX2/4HX10-/
      DATA E/2H20,2H19,2H18,2H17,2H16,2H15,2H14,2H13,2H12,2H11,2H10,1H9,
     X1H8,1H7,1H6,1H5,1H4,1H3,1H2,1H1,1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,
     Y1H8,1H9,2H10,2H11,2H12,2H13,2H14,2H15,2H16,2H17,2H18,2H19,2H20/
      DATA NE/11*2,19*1,11*2/
      DATA YC/20.,19.,18.,17.,16.,15.,14.,13.,12.,11.,10.,9.,8.,7.,6.,5.
     X,4.,3.,2.,1.,0.,1.,2.,3.,4.,5.,6.,7.,8.,9.,10.,11.,12.,13.,14.,15.
     Y  ,16.,17.,18.,19.,20./
      M=3
      IF(ICALL.EQ.1)M=-3
      GO TO (100,200),LXY
C       X AXIS
  100 XLW=X1/D
      XH=XSCALE+XLW
      CALL PLOT(0.,0.,M)
      CALL PLOT(XLW,0.,3)
      CALL PLOT(XH,0.,2)
      CALL PLOT(XLW,0.,3)
      EPS=1.E-6
      XDIF=ABS((X2-X1)*(1.+EPS))
      IL=IFIX(ALOG10(XDIF))
      IF(IL.LT.0)IL=IL-1
      FCTR=10.0**IL
      XTLAB=X2/FCTR
      XBLAB=X1/FCTR
      ITEST=IFIX(XTLAB-XBLAB+EPS)
      DD=0.1*FLOAT(ITEST)
      D1=(XTLAB-XBLAB)/XSCALE
      DDD=DD/D1
      LMAX=1+IFIX((XTLAB-XBLAB+EPS)/DD)
      XL=XLW-DDD
      XLL=XBLAB-DD
      DO 102 L=1,LMAX
      XL=XL+DDD
      XLL=XLL+DD
      CALL SYMBOL(XL,0.,0.1,13,0.0,-1)
      XL1=XL-.15
      CALL NUMBER(XL1,-.15,0.08,XLL,0.,1)
  102 CONTINUE
      CALL PLOT(0.35,-0.4,3)
      XAT=FLOAT(NT)*0.15
      XSTART=XLW+0.5*(XH-XLW-XAT)
      CALL SYMBOL(XSTART,-0.5,0.15,TEXT,0.,NT)
      FA=FLOAT(IABS(IL))
      DO 306 II=1,21
      IF(ABS(FA-YC(II)).LE.EPS)GO TO 307
  306 CONTINUE
  307 S=E(II)
      J=NE(II)
      XAT=0.15*5.
      XSTART=XLW+0.5*(XH-XLW-XAT)
      IF(IL.GE.0)GO TO 308
      TEX3=TEX2
      NEX=4
      GO TO 309
  308 TEX3=TEX1
      NEX=3
  309 CALL SYMBOL(XSTART,-.75,.15,TEX3,0.,NEX)
      XSTART=XSTART+0.15*FLOAT(NEX)
      CALL SYMBOL(XSTART,-.65,.1,S,0.,J)
      RETURN
C     Y AXIS
  200 CALL PLOT(XLW,0.,M)
      CALL PLOT(XLW,YSCALE,2)
      CALL PLOT(0.,0.,3)
      RETURN
      END
```

Program ANISO

Program ANISO computes the reduced source functions $G_n(\mu, q)$ for anisotropic scattering of electrons from a forward-directed point source in an infinite medium. A v3rsion of this program also exists for the point isotropic source case. While this latter version is not listed here, the modifications required to produce this version are minor, and a listing is available from the author.

The routines are

1) ANISO - Main program, performs Gaussian quadrature.

2) Subroutine SETUP - Fills Gauss quadrature ordinate and weight arrays, reads in previous generation G_n array, computes q-values at which G_{n+1} is to be evaluated.

3) Function PROB - Computes scattering probability density.

4) Function SURFTRP - Surface interpolation routine.

5) Subroutine PHIPSI - Computes Hermite polynomials for use by SURFTRP.

6) Subroutine DERIV - Computes partial derivations for use by SURFTRP.

7) Subroutine SPCOFF - Computes spline interpolation coefficients.

Program ANISO - Gaussian quadrature integration of G_n functions for anisotropic scattering

```
      PROGRAM ANISO(INPUT,OUTPUT,TAPE1,TAPE2)
      DIMENSION Q(201),GN(201,201),W(32),XT(32),XR(32),A(32),GN1(32,32),
     1GN2(32,32),VAL(32),PRBR(32),RMT(32,32)
      DIMENSION GNEW(201),SUM(201)
      REAL MU(201)
      DATA EPS/1.E-5/
      READ 1,I1,I2,IANG,IPRNT,N,ETA
    1 FORMAT(5I5,F10.5)
C     IPRNT =0 DO NOT PRINT OUT ANGULAR SOURCE DISTRIBUTION
C     IPRNT =1        PRINT OUT ANGULAR SOURCE DISTRIBUTION
C      IANG =0 COMPUTE G3(Q) ONLY
C      IANG = 1 COMPUTE BOTH G3(Q) AND G3(Q,MU)
      CALL SETUP(W,A,VAL,Q,MU,GN,KMAX,N)
      TOT=0.
      REWIND 2
      DO 500 I=I1,I2
      NN=N-1
      AN=FLOAT(N)
      DO 10 J=1,32
      XR(J)=0.5*((1.+Q(I))+(1.-Q(I))*W(J))
      XT(J)=0.5*((Q(I)-1.)+(1.+Q(I))*W(J))
   10 CONTINUE
      DO 15 JIN=1,32
      R=XR(JIN)
      IF(R.GT.Q(201))R=Q(201)-EPS
      DO 15 JOUT=1,32
      T=XT(JOUT)
      IF(T.GT.Q(201))T=Q(201)-EPS
      GN1(JOUT,JIN)=SURFTRP(Q,Q,GN,201,201,T,R)
      GN2(JIN,JOUT)=SURFTRP(Q,Q,GN,201,201,R,T)
      RMT(JIN,JOUT)=(R-T)**N
   15 CONTINUE
      IF(IANG.EQ.0) GO TO 250
C     MU LOOP
      GNQ=0.
      DO 50 K=1,KMAX
      AMU=MU(K)
      DO 49 JIN=1,32
      R=XR(JIN)
      PRBR(JIN)=PROB(R,AMU,ETA)*(R-Q(I))**NN
   49 CONTINUE
C     OUTER INTEGRAL LOOP
      ANS=0.
      DO 100 JOUT=1,32
      VAL (JOUT)=0.0
      T=XT(JOUT)
      PRBT=PROB(T,AMU,ETA)
      QQ=(Q(I)-T)**NN
C     INNER INTEGRAL LOOP
      DO 110 JIN=1,32
      R=XR(JIN)
      VAL(JOUT)=VAL(JOUT)+0.5*(1.-Q(I))*A(JIN)*
     1(GN1(JOUT,JIN)*PRBT*QQ+GN2(JIN,JOUT)*PRBR(JIN))/RMT(JIN,JOUT)
  110 CONTINUE
      ANS=ANS+0.5*AN*(1.+Q(I))*A(JOUT)*VAL(JOUT)
  100 CONTINUE
      GNQ=GNQ+ANS*.01
      IF(K.EQ.1.OR.K.EQ.KMAX) GNQ=GNQ-ANS*.005
      SUM(K)=GNQ
      GNEW(K)=ANS
   50 CONTINUE
      IF(IPRNT.EQ.1)PRINT 200,(MU(K),GNEW(K),SUM(K),K=1,KMAX)
      IF(IPRNT.EQ.0)PRINT 205,I,SUM(KMAX)
  205 FORMAT(1X,*I=*,I5,1X,*RESULT OF ANGULAR INTEGRATION=*,E12.5)
  201 FORMAT(1X,*Q =*, E12.5)
  250 ANS=0.
      DO 150 JOUT=1,32
      VAL(JOUT)=0.
      T=XT(JOUT)
      QQ=(Q(I)-T)**NN
      DO 150 JIN=1,32
      R=XR(JIN)
      QR=(R-Q(I))**NN
      VAL(JOUT)=VAL(JOUT)+0.5*(1.-Q(I))*A(JIN)*(GN1(JOUT,JIN)*QQ+
     1GN2(JIN,JOUT)*QR)/RMT(JIN,JOUT)
  160 CONTINUE
      ANS=ANS+0.5*AN*(1.+Q(I))*A(JOUT)*VAL(JOUT)
  150 CONTINUE
      FAC=0.01
      IF(I.EQ.1)FAC=0.005
      IF(I.EQ.201)FAC=0.00495
      TOT=TOT+ANS*FAC
      PRINT 202,I,Q(I),ANS,TOT
      WRITE(2)(MU(K),GNEW(K),SUM(K),K=1,KMAX),ANS
  202 FORMAT(1X,*I=*,I5,1X,*Q(I)=*,E12.5,1X,*GN(Q)=*,E12.5,*INTEGRAL=*,E
     112.5)
  200 FORMAT(4(1X,F5.2,2E12.5))
  500 CONTINUE
      STOP
      END

      FUNCTION PROB(A,B,ETA)
      PROB=ETA*(1.+0.5*ETA)*(1.-A*B+ETA)/(ETA**2+2.*ETA*(1.-A*B)+
     1(A-B)**2)**1.5
      RETURN
      END
```

Subroutine SETUP-Fills Gaussian quadrature ordinate and weight arrays, reads in previous generation G_n array, and computes the q array

```
      SUBROUTINE SETUP(W,A,X,Q,MU,GN,KMAX,N)
      DIMENSION W(32),A(32),X(32),Q(201),GN(201,201)
      DIMENSION A1(201),B(201),R(16),S(16)
      REAL MU(201)
      DATA R/         .0483076656877383162,       .1444719615827964935,
     1                .2392873622521370745,       .3318686022821276498,
     2                .4213512761306353454,       .5068999089322293900,
     3                .5877157572407623290,       .6630442669302152009,
     4                .7321821187402896804,       .7944837959679424069,
     5                .8493676137325699701,       .8963211557660521239,
     6                .9349060759377396892,       .9647622555875064308,
     7                .9856115115452683354,       .9972638618494815635/
      DATA S/         .0965400885147278005,       .0956387200792748594,
     1                .0938443990808045656,       .0911738786957638847,
     2                .0876520930044038111,       .0833119242269467552,
     3                .0781938957870703064,       .0723457941088485062,
     4                .0658222227763618468,       .0586840934785355471,
     5                .0509980592623761762,       .0428358980222266806,
     6                .0342738629130214331,       .0253920653092620595,
     7                .0162743947309056706,       .0070186100094700966/
      DO 1 I=1,16
      W(I)=R(I)
      A(I)=S(I)
    1 CONTINUE
      REWIND 1
      IF(N.GT.2)GO TO 10
      DO 100 I=1,201
      READ(1)(A1(J),GN(I,J),B(J),J=1,201),ABC
  100 CONTINUE
      GO TO 20
   10 DO 110 J=1,201
      READ(1)(A1(I),GN(I,J),B(I),I=1,201),ABC
  110 CONTINUE
   20 CONTINUE
      KMAX=201
      MU(1)=-1.
      DO 51 I=2,KMAX
      MU(I)=MU(I-1)+.01
   51 CONTINUE
      DO 2 I=1,16
      X(I)=W(17-I)
    2 X(I+16)=-W(I)
      DO 3 I=1,32
    3 W(I)=X(I)
      DO 4 I=1,16
      X(I)=A(17-I)
    4 X(I+16)=A(I)
      DO 5 I=1,32
    5 A(I)=X(I)
      Q(1)=-1.
      DO 6 I=2,200
    6 Q(I)=Q(I-1)+.01
      Q(201)=0.999
      RETURN
      END
```

Function SURFTRP-Surface interpolation routine

```
      FUNCTION SURFTRP(X,Y,F,IMAX,JMAX,XP,YP)
      DIMENSION X(IMAX),Y(JMAX),F(IMAX,JMAX)
C
C     FUNCTION SURFTRP IS A SURFACE INTERPOLATION PROGRAM.   GIVEN A
C     FUNCTION F OF TWO VARIABLES X AND Y, THE PROGRAM RETURNS AN
C     INTERPOLATED VALUE, SURFTRP, OF THE FUNCTION F(X,Y) AT THE USER
C     SPECIFIED POINT (XP,YP). IN ORDER TO USE THIS FUNCTION PROGRAM,
C     THE CALLING PROGRAM MUST PROVIDE STORAGE VIA A DIMENSION STATEMENT
C     FOR THE ARRAYS X,Y AND F(X,Y).  AN EXAMPLE MIGHT BE---
C       DIMENSION X(100),Y(250),F(100,250)
C
C     AN EXAMPLE OF A CALL TO SURFTRP MIGHT BE ---
C       FN =  SURFTRP(X,Y,F,IMAX,JMAX,XP,YP)  ,
C     WHERE X, Y AND F ARE DEFINED ABOVE,
C          IMAX IS THE LENGTH OF THE ARRAY X,
C          JMAX IS THE LENGTH OF THE ARRAY Y, AND
C          SURFTRP IS THE INTERPOLATED VALUE OF F(X,Y) AT THE POINT
C          (XP,YP).
C     VALUES FOR THE PARAMETERS IMAX AND JMAX MUST BE SPECIFIED IN THE
C     CALLING PROGRAM.
C
C
      IFLAG=0
      SURFTRP=0.
C
C
C     THE INTERPOLATION WEIGHTING FUNCTIONS ARE CUBIC HERMITE POLYNOMIALS,
C     PHI AND PSI.  THE ALGORITHM FOR EVALUATING AND USING THESE IS GIVEN
C     IN P.M. PRENTER, *SPLINES AND VARIATIONAL METHODS*, WILEY INTERSCI-
C     ENCE, PP. 53-57, 130(1975).(THERE IS A CRUCIAL MISPRINT ON PAGE 53
C     OF THIS BOOK. EQN(4) MUST BE MULTIPLIED BY (-1). THIS ERROR IS
C     PROPAGATED THROUGHOUT THE PAGES CITED ABOVE.  ) THE PARTIAL
C     DERIVATIVES OF F(X,Y) WERE EVALUATED USING FIVE POINT NATURAL
C     SPLINES ACCORDING TO THE ALGORITHM GIVEN IN L.F. SHAMPINE AND R.C.
C     ALLEN, *NUMERICAL COMPUTING*, SAUNDERS, PP. 54-57(1973).  THE MIXED
C     SECOND-PARTIAL IS EVALUATED BY CENTRAL DIFFERENCES.
C
C     BRACKET THE LOCATION OF THE POINT XP IN THE X ARRAY. IP IS THE INDEX
C     CORRESPONDING TO THE FIRST POINT IN THE X ARRAY THAT IS LARGER THAN
C     XP.  EVALUATE THE HERMITE POLYNOMIALS PHI AND PSI FOR XP.
C
      CALL PHIPSI(X,IMAX,XP,1,PHIX1,PSIX1,PHIX2,PSIX2,IP,IFLAG)
      IF(IFLAG.EQ.1)RETURN
C
C     BRACKET THE LOCATION OF THE POINT YP IN THE Y ARRAY. JP IS THE INDEX
C     CORRESPONDING TO THE FIRST POINT IN THE Y ARRAY THAT IS LARGER THAN
C     YP.  EVALUATE THE HERMITE POLYNOMIALS PHI AND PSI FOR YP.
C
      CALL PHIPSI(Y,JMAX,YP,2,PHIY1,PSIY1,PHIY2,PSIY2,JP,IFLAG)
      IF(IP.EQ.1.OR.JP.EQ.1.OR.IFLAG.EQ.1)RETURN
C
C
C     EVALUATE THE PARTIAL DERIVATIVES AT THE FOUR POINTS SURROUNDING THE
C     POINT  (XP,YP).
C
C
      II=IP-1
      JJ=JP-1
C     EVALUATE THE PARTIAL DERIVATIVES AT (X(IP),Y(JP))
C
      CALL DERIV(DFDX11,DFDY11,DFXY11,IP,JP,X,Y,F,IMAX,JMAX)
C
C     EVALUATE THE PARTIAL DERIVATIVES AT (X(IP-1),Y(JP))
C
      CALL DERIV(DFDX21,DFDY21,DFXY21,II,JP,X,Y,F,IMAX,JMAX)
C
C     EVALUATE THE PARTIAL DERIVATIVES AT (X(IP-1),Y(JP-1))
C
      CALL DERIV(DFDX22,DFDY22,DFXY22,II,JJ,X,Y,F,IMAX,JMAX)
C
C     EVALUATE THE PARTIAL DERIVATIVES AT (X(IP),Y(JP-1))
C
      CALL DERIV(DFDX12,DFDY12,DFXY12,IP,JJ,X,Y,F,IMAX,JMAX)
C
C
C     COMPUTE INTERPOLATED VALUE F(XP,YP) USING PRENTER ALGORITHM(OP.CIT.)
C
      SURFTPP=F(IP,JP)*PHIX1*PHIY1+F(IP-1,JP)*PHIX2*PHIY1+
     X        F(IP-1,JP-1)*PHIX2*PHIY2+F(IP,JP-1)*PHIX1*PHIY2+
     X        DFDX11*PSIX1*PHIY1+DFDX21*PSIX2*PHIY1+
     X        DFDX22*PSIX2*PHIY2+DFDX12*PSIX1*PHIY2+
     X        DFDY11*PHIX1*PSIY1+DFDY21*PHIX2*PSIY1+
     X        DFDY22*PHIX2*PSIY2+DFDY12*PHIX1*PSIY2+
     X        DFXY11*PSIX1*PSIY1+DFXY21*PSIX2*PSIY1+
     X        DFXY22*PSIX2*PSIY2+DFXY12*PSIX1*PSIY2
C
      RETURN
      END

      SUBROUTINE PHIPSI(Z,KMAX,ZP,KXY,PH1,PS1,PH2,PS2,IP,IFLAG)
C
C     THIS SUBROUTINE COMPUTES THE HERMITE POLYNOMIALS ACCORDING TO THE
C     PRENTER ALGORITHM(OP.CIT.PP.53-57)
C
C
      DIMENSION Z(KMAX)
      IFLAG=0
    8 PH1=PS1=PH2=PS2=0.
      DO 100 I=1,KMAX
      IF(ZP.LT.Z(I))GO TO 104
  100 CONTINUE
      IFLAG=1
  103 IF(KXY.EQ.1)PRINT 102,ZP
      IF(KXY.EQ.2)PRINT 105,ZP
      RETURN
  102 FORMAT(1X,*X VALUE *, E12.5,1X,*OUT OF RANGE*)
  105 FORMAT(1X,*Y VALUE *, E12.5,1X,*OUT OF RANGE*)
  104 IP=I
      IF(IP.EQ.1) GO TO 103
      IP1=IP-1
      A=Z(IP)-Z(IP1)
      T1=((ZP-Z(IP1))/A)**2
      PH1=T1/A*(2.*(Z(IP-)-ZP)+A)
      PS1=T1*(ZP-Z(IP))
      T1=((Z(IP)-ZP)/A)**2
      PH2=-T1/A*(2.*(Z(IP1)-ZP)-A)
      PS2=T1*(ZP-Z(IP1))
      RETURN
      END
      SUBROUTINE DERIV(DFDX,DFDY,DFXY,I,J,X,Y,F,IMAX,JMAX)
C
C     THIS SUBROUTINE COMPUTES THE PARTIAL DERIVATIVES OF F AT EACH OF
C     THE FOUR POINTS SURROUNDING (XP,YP) USING THE SHAMPINE ALGORITHM(OP.
C     CIT. PP. 54-57) USING FIVE POINT NATURAL SPLINES.  THE POINT
C     (X(I),Y(J)) IS TAKEN AS THE CENTRAL (THIRD) POINT.
C
      DIMENSION X(IMAX),Y(JMAX),F(IMAX,JMAX)
      IP1=I+1
      IM1=I-1
      IM2=I-2
      JP1=J+1
      JM1=J-1
      JM2=J-2
      IF(I.GT.2) GO TO 10
C
C     X(I) IS WITHIN TWO POINTS OF THE BEGINNING OF THE CURVE.  SLOPE IS
C     TAKEN AS THAT OF A STRAIGHT LINE PASSING THROUGH X(I) AND X(I+1).
C
      DFDX=(F(IP1,J)-F(I,J))/(X(IP1)-X(I))
      GO TO 100
   10 IF(I.LT.(IMAX-2)) GO TO 20
C
C     X(I) IS WITHIN TWO POINTS OF THE END OF THE CURVE. SLOPE IS TAKEN AS
C     THAT OF A STRAIGHT LINE PASSING THROUGH X(I-1) AND X(I).
C
      DFDX=(F(I,J)-F(IM1,J))/(X(I)-X(IM1))
      GO TO 100
   20 HI=X(IP1)-X(I)
C
C     CALL SPCOFF TO OBTAIN THE SPLINE COEFFICIENTS SM AT X(I) AND SP AT
C     X(I+1).
C
      CALL SPCOFF(IM2,J,SM,SP,1,X,Y,F,IMAX,JMAX)
C
C     EVALUATE PARTIAL DERIVATIVE WITH RESPECT TO X AT X(I) USING
C     SHAMPINE ALGORITHM (OP.CIT.)
C
      DFDX=0.5/HI*(-SM*(X(IP1)-X(I))**2
     1           +2.*(F(IP1,J)-F(I,J)))+HI/6.*(SM-SP)
  100 IF(J.GT.2) GO TO 110
C
C     Y(J) IS WITHIN TWO POINTS OF THE BEGINNING OF THE CURVE.  SLOPE IS
C     TAKEN AS THAT OF A STRAIGHT LINE PASSING THROUGH Y(J) AND Y(J+1).
C
      DFDY=(F(I,JP1)-F(I,J))/(Y(JP1)-Y(J))
      GO TO 200
  110 IF(J.LT.(JMAX-2)) GO TO 120
C
C     Y(J) IS WITHIN TWO POINTS OF THE END OF THE CURVE. SLOPE IS TAKEN AS
C     THAT OF A STRAIGHT LINE PASSING THROUGH Y(J-1) AND Y(J).
C
      DFDY=(F(I,J)-F(I,JM1))/(Y(J)-Y(JM1))
      GO TO 200
  120 HI=Y(JP1)-Y(J)
C
C     CALL SPCOFF TO OBTAIN THE SPLINE COEFFICIENTS SM AT Y(J) AND SP AT
```

```
C     Y(J+1).
C
      CALL SPCOFF(I,JM2,SM,SP,2,X,Y,F,IMAX,JMAX)
C
C     EVALUATE PARTIAL DERIVATIVE WITH RESPECT TO Y AT Y(J) USING
C     SHAMPINE ALGORITHM (OP.CIT.)
C
      DFDY=0.5/HI*(-SM*(Y(JP1)-Y(J))**2
     1     +2.*(F(I,JP1)-F(I,J)))+HI/6.*(SM-SP)
  200 IF(I.EQ.IMAX) IP1=I
      IF(J.EQ.JMAX)JP1=J
      IF(I.EQ.1)IM1=I
      IF(J.EQ.1) JM1=J
C
C     EVALUATE MIXED PARTIAL DERIVATIVE USING CENTRAL DIFFERENCES.
C
      DFXY=(F(IP1,JP1)-F(IP1,JM1)+F(IM1,JM1)-F(IM1,JP1))/
     X     ((X(IP1)-X(IM1))*(Y(JP1)-Y(JM1)))
      RETURN
      END
```

Subroutine SPCOFF is identical to Subroutine SPCOEF used with program GAUSQN (see page 124).

Program GNLEG

Program GNLEG computes the Legendre coefficients of the reduced source functions $G_n(\mu, q)$ for anisotropic scattering of electrons from a forward-directed point source in an infinite medium. A version of this program also exists for the point isotropic source case. The listing for the former will be given here.

The routines are

1) ANISO - Main program - performs Gaussian quadrature integration of Legendre coefficients of G_n.

2) Subroutine DGEN - Computes Legendre coefficients of scattering kernel.

3) Subroutine LEP - Compute Legendre polynomials using recursion formula.

4) Subroutine G2GEN - Computes $G_2(\mu, q)$ function.

5) Subroutine SPCOEF - Computes spline interpolation coefficients.

6) Subroutine NODE - Computes location of Chebyshev modes.

7) Subroutine ECHEB - Evaluates Legendre coefficients of G_n from Chebyshev series representation.

8) Subroutine CHEBY - Evaluates Chebyshev series coefficients for function representation.

9) Subroutine GRULE - Evaluates Gaussian quadrature weights and ordinates.

Program GNLEG- Gaussian quadrature integration of Legendre coefficients of G_n functions

```
      PROGRAM GNLEG(INPUT,OUTPUT,TAPE1,TAPE2)
C
C     THIS PROGRAM COMPUTES THE COEFFICIENTS, GNL(Q), OF THE LEGENDRE
C     EXPANSION OF THE SOURCE FUNCTION, GN(MU,Q), OF THE N-TIMES
C     SCATTERED ELECTRONS. THE CALCULATION, AS FORMULATED BY GARTH,
C     CONSISTS OF AN INTEGRAL EQUATION FOR THE LEGENDRE COEFFICIENTS
C     GNL(Q)  FOR SCATTERING ORDER N IN TERMS OF THE CORRESPONDING
C     LEGENDRE COEFFICIENTS GNM1L(Q) FOR SCATTERING ORDER N-1.   ALL
C     OF THE NUMERICAL INTEGRATIONS ARE PERFORMED USING 16 POINT
C     GAUSSIAN QUADRATURE.
C
      DIMENSION GNM1(201),GNM1L(20,201),PSUBL(20),INDEX(201),S(201),
     1SGL(20,201),QORD1(32),QORD2(32),EMU1(32),EMU2(32),AI(20,20),
     2AJ(20,20),SUM1(20),SUM2(20),GANS(20),Q(201),D(20)
      DIMENSION GNL(20,201),QT(201),NPT(8),UPEDGE(9)
      DIMENSION CNODE(41),ARR(41),CHCOF(41)
      REAL IN(32,20,20),JN(32,20,20),MU(32),WGT(32)
      EQUIVALENCE(EMU1(1),QORD2(1)),(EMU2(1),QORD1(1))
      DATA D,GNM1L,GNL/20*0.,4020*0.,4020*0./
      READ 1,ETA,LMAX,NMIN,NMAX,NORD,NCHSHFT,NCHEB,NQSHIFT
      PRINT 91,ETA,LMAX,NMIN,NMAX,NORD,NCHSHFT,NCHEB,NQSHIFT
   91 FORMAT(1X,*ETA=*,E12.5,1X,*LMAX=*,I10,1X,*NMIN=*,I10,1X,*NMAX=*,I1
     10,1X,*NORD=*,I10/1X,*NCHSHFT=*,I10,1X,*NCHEB=*,I10,1X,*NQSHIFT=*,
     2I10)
    1 FORMAT(F10.0,7I5)
C
C     ETA = RUTHERFORD SCREENING PARAMETER
C     LMAX = MAXIMUM ORDER + 1 OF LEGENDRE EXPANSIONS
C     NMAX = MAXIMUM ORDER OF SCATTERING FOR WHICH THE GNL ARE TO BE
C            COMPUTED
C     NMIN=STARTING ORDER OF SCATTERING FOR WHICH THE GN FUNCTIONS ARE
C          SUPPLIED
C     NORD = ORDER OF GAUSSIAN QUADRATURES
C     NCHSHFT = ORDER OF SCATTERING AT WHICH CHEBYSHEV POLYNOMIAL
C               APPROXIMATION SHIFHTS FROM 41 TO 21 NODES
C     NCHEB = ORDER OF SCATTERING AT WHICH CALCULATION IS SHIFTED FROM
C             LEGENDRE APPROXIMATION ALONE TO LEGENDRE PLUS CHEBYSHEV
C             APPROXIMATIONS
C     NQSHIFT = ORDER OF SCATTERING AT WHICH Q ARRAY IS SHIFTED DOWN
C               FROM 201 POINTS TO 51 POINTS
C
      REWIND 1
      REWIND 2
C
C     SUBROUTINE GRULE COMPUTES THE GAUSSIAN ORDINATES AND WEIGHTS
C
      N1=NORD+1
      NJ=NORD/2
      CALL GRULE(NORD,MU,WGT)
      DO 11 J=1,NJ
      EMU1(J)=-MU(J)
      EMU1(N1-J)=MU(J)
      WGT(N1-J)=WGT(J)
   11 CONTINUE
      DO 12 J=1,NORD
      MU(J)=EMU1(J)
   12 CONTINUE
   70 QT(1)=-1.0
      MP1=MAX+1
      QT(MP1)=1.0
      DO 1000 M=2,MAX
      IF(N.LT.NCHEB)QP=QP+DQ
      IF(N.GE.NCHEB)QP=CNODE(M)
      QT(M)=QP
      FC1=0.5*(1.+QP)
      FC2=0.5*(1.-QP)
C     SELECT THE Q VALUES
      DO 100 J=1,NORD
C      THIS IS FOR THE FIRST MAIN INTEGRAL
      QORD1(J)=FC1+FC2*MU(J)
C     THIS IS FOR THE SECOND MAIN INTEGRAL
      QORD2(J)=-FC2+FC1*MU(J)
  100 CONTINUE
      DO 150 J=1,NORD
      DO 111 L1=1,LMAX
      DO 111 L2=1,LMAX
  111 IN(J,L1,L2)=JN(J,L1,L2)=0.0
      DO 112 K=1,NORD
      CS=EMU1(K)
      CALL LEP(PSUBL,CS,LM1)
      STOR=(((QP-CS)/(QORD1(J)-CS))**(N-1))/(QP-CS)
      DO 112 L1=1,LMAX
      DO 112 L2=1,LMAX
      IN(J,L1,L2)=IN(J,L1,L2)+FC1*WGT(K)*D(L1)*PSUBL(L1)*PSUBL(L2)*STOR
  112 CONTINUE
      DO 115 K=1,NORD
      CS=EMU2(K)
      CALL LEP(PSUBL,CS,LM1)
      STOR=(((CS-QP)/(CS-QORD2(J)))**(N-1))/(CS-QP)
      DO 115 L1=1,LMAX
      DO 115 L2=1,LMAX
      JN(J,L1,L2)=JN(J,L1,L2)+FC2*WGT(K)*D(L1)*PSUBL(L1)*PSUBL(L2)*STOR
  115 CONTINUE
  150 CONTINUE
      DO 120 L2=1,LMAX
      DO 121 I=1,NPTS
      GNM1(I)=GNM1L(L2,I)
  121 S(I)=SGL(L2,I)
      DO 120 L1=1,LMAX
      AI(L1,L2)=AJ(L1,L2)=0.
      DO 122 J=1,NORD
      Q1=QORD1(J)
      Q2=QORD2(J)
      SMG1=SPLINE(NPTS,Q,GNM1,S,INDEX,Q1)
      SMG2=SPLINE(NPTS,Q,GNM1,S,INDEX,Q2)
      AI(L1,L2)=AI(L1,L2)+FC2*WGT(J)*SMG1*IN(J,L1,L2)
      AJ(L1,L2)=AJ(L1,L2)+FC1*WGT(J)*SMG2*JN(J,L1,L2)
  122 CONTINUE
  120 CONTINUE
      DO 130 L1=1,LMAX
      SUM1(L1)=SUM2(L1)=0.0
      DO 136 L2=1,LMAX
      A=0.5+FLOAT(L2-1)
      SUM1(L1)=SUM1(L1)+A*AI(L1,L2)
      GNM1L(LL,I)=GNM1L(LL,I)/TOTAL
  549 CONTINUE
  559 CONTINUE
      DO 560 K=1,201
      GNL(1,K)=GNL(2,K)=GNL(3,K)=0.0
  560 CONTINUE
      GO TO 570
  556 REWIND 1
      DO 2 J=1,NORD
      CS=MJ(J)
      IF(NMIN.EQ.3)CALL G2GEN(ETA,Q,CS,GNM1)
      CALL LEP(PSUBL,CS,LM1)
      DO 21 LL=1,LMAX
      DO 211 I=1,201
      GNM1L(LL,I)=GNM1L(LL,I)+PSUBL(LL)*GNM1(I)*WGT(J)
  211 CONTINUE
   21 CONTINUE
    2 CONTINUE
  570 IF(NQSHIFT.EQ.(NMIN+1))NQSHIFT=NMIN+2
      NPTS=201
      KCHEB=41
      DO 590 N=NMIN,NMAX
      AN=FLOAT(N-1)
      IF(N.NE.(NMIN+1))GO TO 571
      Q(1)=-1.0
      DO 569 I=2,201
      Q(I)=Q(I-1)+0.01
  569 CONTINUE
  571 IF(N.NE.NQSHIFT) GO TO 175
      NPT1=NPTS
      NPTS=51
      DQ=2./FLOAT(NPTS-1)
      DO 71 I=2,NPTS
   71 Q(I)=Q(I-1)+DQ
      DO 40 LL=1,LMAX
      I=0
      DO 40 II=1,NPT1,4
      I=I+1
      GNM1L(LL,I)=GNL(LL,II)
   40 CONTINUE
      Q(NPTS)=1.
  175 DO 4 LL=1,LMAX
      DO 41 I=1,NPTS
   41 GNM1(I)=GNM1L(LL,I)
      CALL SPCOFF(NPTS,Q,GNM1,S,INDEX)
      DO 411 I=1,NPTS
  411 SGL(LL,I)=S(I)
    4 CONTINUE
C     SGL ARE THE SPLINE COEFFICIENTS OF G2L
      QP=-1.0
      MAX=NPTS-1
      IF(N.LT.NCHEB) GO TO 70
      KCHEB=41
      IF(N.GE.NCHSHFT)KCHEB=21
      MAX=KCHEB-1
      CALL NODE(KCHEB,CNODE)
      PRINT 993,(M,CNODE(M),M=1,KCHEB)
      C
      C     SUBROUTINE DGEN COMPUTES THE LEGENDRE COEFFICIENTS OF THE SCREENED
      C     RUTHERFORD SCATTERING KERNEL
      C
            CALL DGEN(ETA,LMAX,D)
            PRINT 501,(J,MU(J),WGT(J),J=1,NORD)
        501 FORMAT(1X,I5,2E16.9)
            LM1=LMAX-1
            PRINT 10,ETA,LM1,(D(L),L=1,LMAX)
         10 FORMAT(1X,*RUTHERFORD SCREENING FACTOR =*,E12.5,/1X,*ORDER OF LEGE
           1NDRE SERIES CALCULATION IS*,I5/1X,*SCATTERING COEFFICIENTS=*,/1X,1
           20E12.5/1X,10E12.5)
      C     OBTAIN THE LEGENDRE COEFFICIENTS OF GNL IN AN ARRAY 101 LONG ON Q.
      C     THEN SPLINE IT FOR USE IN THE MAIN INTEGRAL EQUATION.
      C
      C     GET GNM1(MU,Q) FOR THE 16 MU VALUES AND 101 Q-VALUES.
      C     INTEGRATE DIRECTLY TO GET GNM1L(Q).
            DQ=.01
            Q(1)=-1.0
            DO 3 I=2,200
            Q(I)=Q(I-1)+DQ
          3 CONTINUE
            Q(201)=0.999
            IF(NMIN.EQ.3) GO TO 556
            READ 5,NINT,(NPT(I),I=1,NINT)
            READ 6,(UPEDGE(I),I=1,NINT)
          5 FORMAT(8I10)
          6 FORMAT(8F10.0)
            IK=1
            IL=2
            DOWN=-1.0
            DO 7 NI=1,NINT
            IK=IK+NPT(NI)
            UP=UPEDGE(NI)
            DELQ=(UP-DOWN)/FLOAT(NPT(NI))
            DO 8 I=IL,IK
            Q(I)=Q(I-1)+DELQ
          8 CONTINUE
            IL=IK+1
            DOWN=UP
          7 CONTINUE
            TOTAL=0.0
            DO 555 I=1,201
            READ(1)(GNL(1,K),GNL(2,K),GNL(3,K),K=1,201),ABC
            IF(I.NE.1)TOTAL=TOTAL+ABC*(Q(I)-Q(I-1))
            CS=-1.0
            DO 557 K=1,201
            CALL LEP(PSUBL,CS,LM1)
            DO 558 LL=1,LMAX
            GNM1L(LL,I)=GNM1L(LL,I)+PSUBL(LL)*GNL(2,K)*0.01
        558 CONTINUE
            CS=CS+.01
        557 CONTINUE
        555 CONTINUE
            DO 559 LL=1,LMAX
            GNM1L(LL,1)=GNM1L(LL,201)=0.0
            DO 549 I=2,200
```

```
  136 SUM2(L1)=SUM2(L1)+A*AJ(L1,L2)
      GANS(L1)=AN*(SUM1(L1)+SUM2(L1))
      GNL(L1,M)=GANS(L1)
  130 CONTINUE
 1000 CONTINUE
      DO 1001 L=1,LMAX
 1001 GNL(L,1)=GNL(L,MP1)=0.0
      TOTAL =0.
      IF(MAX.LE.40) GO TO 75
      DO 74 M=2,MAX
      TOTAL=TOTAL+GNL(1,M)*DQ
   74 CONTINUE
      PRINT 159,N
      PRINT 141,TOTAL
      DO 76  LL=1,LMAX
      DO 78 MM=1,NPTS
      GNL(LL,MM)=GNL(LL,MM)/TOTAL
   78 CONTINUE
      WRITE(2)N,LL,NPTS,(GNL(LL,MM),MM=1,NPTS)
   76 CONTINUE
      GO TO 77
   75 DO 1002 L=1,LMAX
      DO 1003 M=1,KCHEB
      ARR(M)=GNL(L,M)
 1003 CONTINUE
      CALL CHEBY(ARR,CNODE,KCHEB,CHCOF)
      PRINT 998,(CHCOF(KK),KK=1,KCHEB)
      QQ=-1.
      DO 1004 J=1,NPTS
      CALL ECHEB(QQ,CHCOF,KCHEB,VAL)
      GNL(L,J)=VAL
      IF(L.EQ.1)TOTAL=TOTAL+DQ*VAL
      QQ=QQ+DQ
 1004 CONTINUE
      DO 550 KK=1,KCHEB
  550 CHCOF(KK)=CHCOF(KK)/TOTAL
      WRITE(2)N,L,KCHEB,(CHCOF(KK),KK=1,KCHEB)
      QQ=-1.
      DO 1005 J=1,NPTS
      CALL ECHEB(QQ,CHCOF,KCHEB,VAL)
      GNL(L,J)=VAL
      QQ=QQ+DQ
 1005 CONTINUE
 1002 CONTINUE
  999 FORMAT(1X,*CHEBYSHEV NODES*/(I5,E16.9))
  998 FORMAT(1X,*CHEBYSHEVCOEFFICIENTS OF GN*/(5E16.9))
      PRINT 159,N
      PRINT 141,TOTAL
   77 PRINT 551
  551 FORMAT(1X,*RENORMALIZED*)
      PRINT 161,(MM,QT(MM),GNL(1,MM),MM=1,NPTS)
  141 FORMAT(1X,*INTEGRAL =*,E12.5)
      DO 585 LL=1,LMAX
      DO 585 I=1,NPTS
      GNM1L(LL,I)=GNL(LL,I)
  585 CONTINUE
  500 CONTINUE


  159 FORMAT(1X,*N=*,I5,2X,*GN(Q)*)
  161 FORMAT(5(I6,F5.2,E12.5))
      STOP
      END
```

Subroutine DGEN- Computes Legendre coefficients of scattering kernel

```
      SUBROUTINE DGEN(ETA,LMAX,D)
      DIMENSION D(20),Z(20)
      IF(LMAX.GT.1.OR.ETA.LT.1.E5)GO TO 5
      D(1)=1.0
      RETURN
    5 CONTINUE
      E1=(1./(1.+0.5*ETA))
      Z(1)=0.0
      Z(2)=ALOG(1.+2./ETA)-E1
      C1=E1/ETA
      TEST=1.E6
      DO 1 L=3,LMAX
      AL=FLOAT(L-2)
      BL=2.*AL+1.
      CL=AL+1.
      Z(L)=(BL*(1.+ETA)*Z(L-1)-CL*Z(L-2)-BL*E1)/AL
      DD=ABS(Z(L)-C1)
      IF(DD.GT.TEST) GO TO 1
      TEST=DD
      LM=L
    1 CONTINUE
      IF(LM.EQ.LMAX) GO TO 3
      DO 2 L=LM,LMAX
      Z(L)=C1
    2 CONTINUE
    3 DO 4 L=1,LMAX
      D(L)=1.-Z(L)/C1
    4 CONTINUE
      RETURN
      END
```

Subroutine LEP is identical to Subroutine LEP used with program PLMETHD. (see page 106).

Subroutine G2GEN - computes the G_2 function

```
      SUBROUTINE G2GEN(ETA,XOS,CS,EXC)
      DIMENSION XOS(201),EXC(201)
      REAL I1(201),I2(201),I3(201),I4(201),I5(201),I6(201)
      IF(ETA.LT.1.E5) GO TO 5
      DO 10 I=1,201
      EXC(I)=0.25*ALOG(2./(1.-XOS(I)))
   10 CONTINUE
      RETURN
    5 CONTINUE
      Y=1./(1.-CS)
      FAC=(Y*(1.+0.5*ETA)/ETA)**2
      ALPHA=(1.+ETA*Y)**2
      BETA=2.*Y*(ETA*Y-ETA-1.)
      GAMMA=Y*Y
      S=ALPHA/ETA**2
      T=BETA/ETA-2.*ALPHA/ETA**2
      U=GAMMA-S-T
      DISC=4.*U*S-T*T
      AA=S
      BB=0.5*(T+2.0*S)
      CC=S+T+U
      C1=AA*CC-BB*BB
      C2=5.*BB*CC-6.*BB**3/AA
      C3=3.*BB/(AA**2.5)
      C4=2.*CC*CC-3.*BB*BB*CC/AA
      EXC(1)=0.
      B1=B2=B3=B4=B5=0.
      DO 200 I=1,201
      IF(ETA.LT.0.1) GO TO 199
      X=1.+ETA/(1.-XOS(I))
      RADCL=SQRT(S*X**2+T*X+U)
      EPS=ALOG(2.*U+T*X+2.*SQRT(U)*RADCL)-ALOG(X)
      I1(I)   =-2.*(2.*U+T*X)/(DISC*RADCL)
      I2(I)   =2.*(2.*S*X+T)/(DISC*RADCL)
      I3(I)   =-2.*(S*T*X-2.*U*S+T*T)/(U*DISC*RADCL)-EPS/(U**1.5)
      I4(I)   =-1./(U*X*RADCL)-3.*T/(2.*U)*I3(I) --2.*S/U*I2(I)
      I5(I)   =((-4.*U*S+2.*T*T)*X+2.*U*T)/(S*DISC*RADCL)+
     1(ALOG(2.*SQRT(S)*RADCL+2.*S*X+T))/S**1.5
      IF(I.EQ.1) GO TO 70
      B1=B1+I1(I)-I1(I-1)
      B2=B2+I2(I)-I2(I-1)
      B3=B3+I3(I)-I3(I-1)
      B4=B4+I4(I)-I4(I-1)
      B5=B5+I5(I)-I5(I-1)
   70 EXC(I)= FAC*(-(1.+3.*Y+4./ETA)*B1+3.*B2*(1.+Y+2./ETA)-B3*(Y+3.+4./
     1ETA)+B4*(1.+1./ETA)+B5*(Y+1./ETA))
      GO TO 200
  199 Z=ETA/(1.-XOS(I))
      RADCL=SQRT(AA*Z*Z+2.*BB*Z+CC)
      TERM1=C2/C1*Z+C4/C1
      TERM2=ALOG(RADCL+(AA*Z+BB)/SQRT(AA))
      I6(I)=1./(AA*RADCL)*(Z*Z+TERM1)-C3*TERM2
      DEL=4.*(AA*CC-BB*BB)
      AJ2=(-(DEL-4.*BB*BB)*Z+4.*CC*BB)/(AA*DEL*RADCL)+1./(AA**1.5)*
     1ALOG(2.*SQRT(AA)*RADCL+2.*AA*Z+2.*BB)
      A6=1./(2.*AA)*(-5.*BB*I6(I)+Z**3/RADCL-3.*CC*AJ2)
      I6(I)=A6*(1./ETA+2.-Y)+I6(I)
      IF(I.EQ.1) GO TO 200
      EXC(I)=EXC(I-1)+FAC*(I6(I)-I6(I-1))
  200 CONTINUE
      RETURN
      END
```

Subroutine SPCOEF and Function SPLINE are identical to those used with Program GAUSQN (see pages 124, 125).

Subroutines ECHEB and CHEBY are taken from R. Broucke, "Ten Subroutines for the Manipulation of Chebyshev Series", Alg. 446, Comm. ACM, 16, No. 4 (1973)

```
      SUBROUTINE ECHEB(X,COEF,NPL,FX)
      DIMENSION COEF(1)
      BR=BRPP=0.
      DO 10 K=1,NPL
      J=NPL-K+1
      BRP2=BRPP
      BRPP=BR
      BR=2.*X*BRPP-BRP2+COEF(J)
   10 CONTINUE
      FX=0.5*(BR-BRP2)
      RETURN
      END
```

```
      SUBROUTINE CHEBY(F,T,NP1,AR)
      DIMENSION F(1),T(1),AR(1)
      F(NP1)=F(NP1)/2.
      DO 100 IR=1,NP1
      Z=T(IR)
      TZ=Z+Z
      B1=B2=0.0
      NP=NP1-1
      DN=NP
      DO 15 J=1,NP
      K=NP1-J+1
      TEMP=B1
      B1=TZ*B1-B2+F(K)
      B2=TEMP
   15 CONTINUE
      ANS=Z*B1-B2+F(1)/2.
      AR(IR)=(ANS+ANS)/DN
  100 CONTINUE
      RETURN
      END
```

Subroutine NODE was extracted from a set of algorithms by D.E. Amos and S.L. Daniel, "CDC 6600 Utility Routines for Chebyshev Approximation and Function Inversion", Sandia Laboratories Report SC-DR-72 0917(1972).

```
      SUBROUTINE NODE(NCHEB,X)
      DIMENSION X(41),T(41)
      PI=3.1415926535898
      APB=0.
      BMA=2.
      N=NCHEB-1
      DN=N
      THET=PI/DN
      T(1)=1.0
      T(2)=COS(THET)
      TONE=T(2)
      VONE=SIN(THET)
      VK=VONE
C
      DO 5 K=2,N
      TK=T(K)
      T(K+1)=TONE*TK-VK*VONE
      VK=VK*TONE+TK*VONE
      X(K)=(TK*BMA+APB)/2.
    5 CONTINUE
      X(1)=(T(1)*BMA+APB)/2.
      X(NCHEB)=(T(NCHEB)*BMA+APB)/2.
      RETURN
      END
```

Subroutine GRULE- Evaluates Gaussian quadrature weights and coefficients

```
      SUBROUTINE GRULE(N,X,W)
C
C     THIS SUBROUTINE WAS FOUND ON PAGE 369 OF DAVIS AND RABINOWITZ,
C     "METHODS OF NUMERICAL INTEGRATION"ACADEMIC PRESS,N.Y.(1957)
C
C     IT CALCULATES GAUSSIAN WEIGHTS AND ABSCISSAS USING NEWTON'S
C     METHOD FOR FINDING ROOTS GIVEN A GOOD FIRST ESTIMATE.
C     MODIFIED 4 NOVEMBER 1977 BY J.C.GARTH TO ITERATE ONCE FOR
C     FINDING MORE ACCURATE VALUES FOR THE ROOTS.  THE DOUBLE PRECISION
C     VERSION GIVES AGREEMENT WITH THE TABLES IN ABRAMOWITZ ET AL.
C     TO 15 DECIMAL PLACES.  THE SINGLE PRECISION ROUTINE IS GOOD TO
C     ABOUT TWELVE PLACES.
C
      DIMENSION X(1),W(1)
C
      M=(N+1)/2
      E1=N*(N+1)
      DO 1 I=1,M
      T=(4*I-1)*3.1415926536/(4*N+2)
      X0=(1.-(1.-1./N)/(8.*N*N))*COS(T)
      J=0
    2 CONTINUE
      J=J+1
      PKM1=1.
      PK=X0
      DO 3 K=2,N
      T1=X0*PK
      PKP1=T1-PKM1-(T1-PKM1)/K+T1
      PKM1=PK
    3 PK=PKP1
      DEN=1.-X0*X0
      D1=N*(PKM1-X0*PK)
      DPN=D1/DEN
      D2PN=(2.*X0*DPN-E1*PK)/DEN
      D3PN=(4.*X0*D2PN+(2.-E1)*DPN)/DEN
      D4PN=(6.*X0*D3PN+(6.-E1)*D2PN)/DEN
      U=PK/DPN
      V=D2PN/DPN
      H=-U*(1.+.5*U*(V+U*(V*V-U*D3PN/(3.*DPN))))
      P=PK+H*(DPN+.5*H*(D2PN+H/3.*(D3PN+.25*H*D4PN)))
      DP=DPN+H*(DPN+.5*H*(D3PN+H*D4PN/3.))
      H=H-P/DP
C     J IS THE NUMBER OF ITERATIONS USED.
      IF (J.GE.2) GO TO 4
      X0=X0+H
      GO TO 2
    4 CONTINUE
      X(I)=X0+H
      FX=D1-H*E1*(PK+.5*H*(DPN+H/3.*(D2PN*.25*H*(D3PN+.2*H*D4PN))))
    1 W(I)=2.*(1.-X(I)*X(I))/(FX*FX)
      IF ((M+M).GT.N) X(M)=0.
      RETURN
      END
```

Program LOOK

Program LOOK reads the files, stored on magnetic tape, of Legendre coefficients of the reduced source functions and prints the data in an easily legible form. A listing of the program is given here along with two sample pages of output.

```
      PROGRAM LOOK(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3)
      DIMENSION GN(10,201),Q(201),IOPT(3),LMX(3)
      DATA LMX/5,8/
      READ 1,(IOPT(I),I=1,3)
C
C  IOPT(1) .NE. 0 , PRINT DATA FOR ETA = 10.0
C  IOPT(2) .NE. 0 , PRINT DATA FOR ETA = 1.0
C  IOPT(3) WILL BE USED FOR ETA=0.1
C
    1 FORMAT(3I5)
      IF(IOPT(1).EQ.0)GO TO 200
C
C  LMAX = MAXIMUM LEGENDRE COEFFICIENT ORDER
C         (=4 FOR ETA = 10.0,  =7 FOR ETA=1.0)
C
  100 I=1
      LMAX=LMX(I)
C
C  Q IS EQUISPACED FOR ETA = 10.0
C
      Q(1)=-1.0
      DO 101 J=2,201
  101 Q(J)=Q(J-1)+.01
C
C  N IS THE ORDER OF SCATTERING
C
      DO 102 N=3,10
C
C  L IS THE INDEX FOR THE LEGENDRE COEFFICIENT ORDER
C
C  M IS THE INDEX FOR THE Q ARRAY
C
      READ(1)((GN(L,M),L=1,LMAX),M=1,201)
      CALL PRNTR(N,I,Q,GN,LMAX)
  102 CONTINUE
  200 IF(IOPT(2).EQ.0)GO TO 300
      I=2
      LMAX=LMX(I)
C
C  N IS THE ORDER OF SCATTERING
C
      DO 201 N=3,10
C
C  M IS THE INDEX FOR THE Q ARRAY
C
      DO 202 M=1,201
C
C  L IS THE INDEX FOR THE LEGENDRE COEFFICIENT ORDER
C
C  Q IS NOT EQUISPACED FOR ETA=1.0.   IT IS READ IN.
C
      READ(2)Q(M),(GN(L,M),L=1,LMAX)
  202 CONTINUE
      CALL PRNTR(N,I,Q,GN,LMAX)
  201 CONTINUE
  300 STOP
      END
      SUBROUTINE PRNTR(N,I,Q,GN,LMAX)
      DIMENSION GN(10,201),ETA(3),Q(201)
      DATA ETA/10.,1.0,0.1/
      PRINT 100,ETA(I),N
  100 FORMAT(*1*,47X,*RUTHERFORD SCREENING FACTOR, ETA=*,1X,F5.2,/48X,*O
     1RDER OF SCATTERING =*,I5,/48X,*LEGENDRE COEFFICIENTS OF GN*)
      DO 10 L=1,LMAX
      LL=L-1
      PRINT 101,LL
  101 FORMAT(48X,*LEGENDRE SERIES ORDER =*,I5)
      PRINT 102
  102 FORMAT(4X,*J*,4X,*Q*,9X,*GN*,8X,*J*,4X,*Q*,8X,*GN*,8X,*J*,4X,*Q*,8
     1X,*GN*,8X,*J*,4X,*Q*,8X,*GN*,8X,*J*,4X,*Q*,8X,*GN*)
      PRINT 103,(J,Q(J),GN(L,J),J=1,201)
  103 FORMAT(5(I6,F6.3,E12.5))
   10 CONTINUE
      RETURN
      END
```

```
                                      RUTHERFORD SCREENING FACTOR, ETA= 10.00
                                      ORDER OF SCATTERING =     3
                                      LEGENDRE COEFFICIENTS OF GN
                                      LEGENDRE SERIES ORDER =     0
  J     Q          GN          J     Q          GN          J     Q          GN          J     Q          GN          J     Q          GN
  1-1.000 0.                   2 -.990  .16236E-03      3 -.980  .59640E-03      4 -.970  .12334E-02      5 -.960  .20829E-02
  6 -.950  .31208E-02          7 -.940  .43350E-02      8 -.930  .57138E-02      9 -.920  .72646E-02     10 -.910  .89646E-02
 11 -.900  .10814E-01         12 -.890  .12807E-01     13 -.880  .14940E-01     14 -.870  .17208E-01     15 -.860  .19509E-01
 16 -.850  .22136E-01         17 -.840  .24702E-01     18 -.830  .27563E-01     19 -.820  .30464E-01     20 -.810  .33477E-01
 21 -.800  .36604E-01         22 -.790  .39844E-01     23 -.780  .43193E-01     24 -.770  .46650E-01     25 -.760  .50212E-01
 26 -.750  .53879E-01         27 -.740  .57647E-01     28 -.730  .61515E-01     29 -.720  .65482E-01     30 -.710  .69546E-01
 31 -.700  .73705E-01         32 -.690  .77957E-01     33 -.680  .82302E-01     34 -.670  .86738E-01     35 -.660  .91262E-01
 36 -.650  .95876E-01         37 -.640  .10058E+00     38 -.630  .10536E+00     39 -.620  .11023E+00     40 -.610  .11516E+00
 41 -.600  .12022E+00         42 -.590  .12533E+00     43 -.580  .13053E+00     44 -.570  .13580E+00     45 -.560  .14115E+00
 46 -.550  .14657E+00         47 -.540  .15208E+00     48 -.530  .15766E+00     49 -.520  .16330E+00     50 -.510  .16902E+00
 51 -.500  .17481E+00         52 -.490  .18067E+00     53 -.480  .18660E+00     54 -.470  .19260E+00     55 -.460  .19867E+00
 56 -.450  .20480E+00         57 -.440  .21100E+00     58 -.430  .21726E+00     59 -.420  .22358E+00     60 -.410  .22997E+00
 61 -.400  .23642E+00         62 -.390  .24293E+00     63 -.380  .24950E+00     64 -.370  .25613E+00     65 -.360  .26281E+00
 66 -.350  .26956E+00         67 -.340  .27635E+00     68 -.330  .28321E+00     69 -.320  .29012E+00     70 -.310  .29708E+00
 71 -.300  .30409E+00         72 -.290  .31116E+00     73 -.280  .31827E+00     74 -.270  .32544E+00     75 -.260  .33265E+00
 76 -.250  .33991E+00         77 -.240  .34722E+00     78 -.230  .35458E+00     79 -.220  .36197E+00     80 -.210  .36942E+00
 81 -.200  .37690E+00         82 -.190  .38443E+00     83 -.180  .39199E+00     84 -.170  .39960E+00     85 -.160  .40724E+00
 86 -.150  .41493E+00         87 -.140  .42265E+00     88 -.130  .43040E+00     89 -.120  .43819E+00     90 -.110  .44601E+00
 91 -.100  .45386E+00         92 -.090  .46175E+00     93 -.080  .46966E+00     94 -.070  .47761E+00     95 -.060  .48558E+00
 96 -.050  .49357E+00         97 -.040  .50160E+00     98 -.030  .50964E+00     99 -.020  .51771E+00    100 -.010  .52580E+00
101  .000  .53390E+00        102  .010  .54203E+00    103  .020  .55017E+00    104  .030  .55833E+00    105  .040  .56651E+00
106  .050  .57469E+00        107  .060  .58289E+00    108  .070  .59109E+00    109  .080  .59931E+00    110  .090  .60752E+00
111  .100  .61575E+00        112  .110  .62397E+00    113  .120  .63220E+00    114  .130  .64042E+00    115  .140  .64864E+00
116  .150  .65685E+00        117  .160  .66505E+00    118  .170  .67325E+00    119  .180  .68143E+00    120  .190  .68959E+00
121  .200  .69774E+00        122  .210  .70586E+00    123  .220  .71396E+00    124  .230  .72204E+00    125  .240  .73009E+00
126  .250  .73810E+00        127  .260  .74608E+00    128  .270  .75402E+00    129  .280  .76192E+00    130  .290  .76977E+00
131  .300  .77758E+00        132  .310  .78533E+00    133  .320  .79303E+00    134  .330  .80067E+00    135  .340  .80825E+00
136  .350  .81575E+00        137  .360  .82319E+00    138  .370  .83055E+00    139  .380  .83782E+00    140  .390  .84501E+00
141  .400  .85210E+00        142  .410  .85910E+00    143  .420  .86599E+00    144  .430  .87277E+00    145  .440  .87944E+00
146  .450  .88598E+00        147  .460  .89239E+00    148  .470  .89866E+00    149  .480  .90479E+00    150  .490  .91076E+00
151  .500  .91657E+00        152  .510  .92221E+00    153  .520  .92767E+00    154  .530  .93293E+00    155  .540  .93800E+00
156  .550  .94284E+00        157  .560  .94747E+00    158  .570  .95185E+00    159  .580  .95598E+00    160  .590  .95984E+00
161  .600  .96342E+00        162  .610  .96670E+00    163  .620  .96966E+00    164  .630  .97228E+00    165  .640  .97454E+00
166  .650  .97643E+00        167  .660  .97791E+00    168  .670  .97897E+00    169  .680  .97958E+00    170  .690  .97971E+00
171  .700  .97933E+00        172  .710  .97841E+00    173  .720  .97690E+00    174  .730  .97477E+00    175  .740  .97198E+00
176  .750  .96844E+00        177  .760  .96413E+00    178  .770  .95900E+00    179  .780  .95298E+00    180  .790  .94600E+00
181  .800  .93798E+00        182  .810  .92890E+00    183  .820  .91838E+00    184  .830  .90660E+00    185  .840  .89334E+00
186  .850  .87843E+00        187  .860  .86169E+00    188  .870  .84292E+00    189  .880  .82188E+00    190  .890  .79826E+00
191  .900  .77170E+00        192  .910  .74176E+00    193  .920  .70786E+00    194  .930  .66933E+00    195  .940  .62514E+00
196  .950  .57396E+00        197  .960  .51374E+00    198  .970  .44159E+00    199  .980  .35065E+00    200  .990  .23233E+00
201 1.000 0.
                                      LEGENDRE SERIES ORDER =     1
  J     Q          GN          J     Q          GN          J     Q          GN          J     Q          GN          J     Q          GN
  1-1.000 0.                   2 -.990 -.85595E-05      3 -.980 -.30310E-04      4 -.970 -.62769E-04      5 -.960 -.10458E-03
  6 -.950 -.15476E-03          7 -.940 -.21257E-03      8 -.930 -.27737E-03      9 -.920 -.34862E-03     10 -.910 -.42584E-03
 11 -.900 -.50863E-03         12 -.890 -.59659E-03     13 -.880 -.68939E-03     14 -.870 -.78667E-03     15 -.860 -.88817E-03
 16 -.850 -.99359E-03         17 -.840 -.11027E-02     18 -.830 -.12152E-02     19 -.820 -.13309E-02     20 -.810 -.14495E-02
 21 -.800 -.15709E-02         22 -.790 -.16948E-02     23 -.780 -.18211E-02     24 -.770 -.19495E-02     25 -.760 -.20799E-02
 26 -.750 -.22121E-02         27 -.740 -.23459E-02     28 -.730 -.24812E-02     29 -.720 -.26177E-02     30 -.710 -.27555E-02
 31 -.700 -.28942E-02         32 -.690 -.30338E-02     33 -.680 -.31741E-02     34 -.670 -.33150E-02     35 -.660 -.34563E-02
 36 -.650 -.35979E-02         37 -.640 -.37397E-02     38 -.630 -.38815E-02     39 -.620 -.40233E-02     40 -.610 -.41649E-02
 41 -.600 -.43061E-02         42 -.590 -.44469E-02     43 -.580 -.45871E-02     44 -.570 -.47266E-02     45 -.560 -.48654E-02
 46 -.550 -.50032E-02         47 -.540 -.51401E-02     48 -.530 -.52759E-02     49 -.520 -.54103E-02     50 -.510 -.55435E-02
 51 -.500 -.56752E-02         52 -.490 -.58054E-02     53 -.480 -.59339E-02     54 -.470 -.60607E-02     55 -.460 -.61856E-02
 56 -.450 -.63086E-02         57 -.440 -.64296E-02     58 -.430 -.65484E-02     59 -.420 -.66650E-02     60 -.410 -.67792E-02
 61 -.400 -.68910E-02         62 -.390 -.70002E-02     63 -.380 -.71069E-02     64 -.370 -.72108E-02     65 -.360 -.73119E-02
 71 -.300 -.78544E-02     72 -.290 -.79324E-02     73 -.280 -.80055E-02     74 -.270 -.80302E-02     75 -.260 -.81482E-02
 76 -.250 -.82123E-02     77 -.240 -.82726E-02     78 -.230 -.83289E-02     79 -.220 -.83811E-02     80 -.210 -.84291E-02
 81 -.200 -.84729E-02     82 -.190 -.85123E-02     83 -.180 -.85473E-02     84 -.170 -.85778E-02     85 -.160 -.86037E-02
 86 -.150 -.86250E-02     87 -.140 -.86414E-02     88 -.130 -.86530E-02     89 -.120 -.86597E-02     90 -.110 -.86614E-02
 91 -.100 -.86579E-02     92 -.090 -.86493E-02     93 -.080 -.86354E-02     94 -.070 -.86161E-02     95 -.060 -.85914E-02
 96 -.050 -.85611E-02     97 -.040 -.85253E-02     98 -.030 -.84838E-02     99 -.020 -.84365E-02    100 -.010 -.83834E-02
101  .000 -.83243E-02    102  .010 -.82593E-02    103  .020 -.81881E-02    104  .030 -.81108E-02    105  .040 -.80273E-02
106  .050 -.79374E-02    107  .060 -.78411E-02    108  .070 -.77383E-02    109  .080 -.76289E-02    110  .090 -.75129E-02
111  .100 -.73902E-02    112  .110 -.72607E-02    113  .120 -.71243E-02    114  .130 -.69809E-02    115  .140 -.68305E-02
116  .150 -.66730E-02    117  .160 -.65083E-02    118  .170 -.63363E-02    119  .180 -.61569E-02    120  .190 -.59702E-02
121  .200 -.57760E-02    122  .210 -.55742E-02    123  .220 -.53647E-02    124  .230 -.51476E-02    125  .240 -.49227E-02
126  .250 -.46900E-02    127  .260 -.44494E-02    128  .270 -.42008E-02    129  .280 -.39444E-02    130  .290 -.36798E-02
131  .300 -.34072E-02    132  .310 -.31264E-02    133  .320 -.28374E-02    134  .330 -.25403E-02    135  .340 -.22348E-02
136  .350 -.19211E-02    137  .360 -.15990E-02    138  .370 -.12686E-02    139  .380 -.92980E-03    140  .390 -.58260E-03
141  .400 -.22700E-03    142  .410  .13701E-03    143  .420  .50942E-03    144  .430  .89020E-03    145  .440  .12794E-02
146  .450  .16768E-02    147  .460  .20826E-02    148  .470  .24967E-02    149  .480  .29190E-02    150  .490  .33494E-02
151  .500  .37879E-02    152  .510  .42344E-02    153  .520  .46887E-02    154  .530  .51507E-02    155  .540  .56203E-02
156  .550  .60974E-02    157  .560  .65817E-02    158  .570  .70730E-02    159  .580  .75711E-02    160  .590  .80758E-02
161  .600  .85867E-02    162  .610  .91036E-02    163  .620  .96260E-02    164  .630  .10153E-01    165  .640  .10686E-01
166  .650  .11222E-01    167  .660  .11762E-01    168  .670  .12305E-01    169  .680  .12850E-01    170  .690  .13397E-01
171  .700  .13944E-01    172  .710  .14491E-01    173  .720  .15036E-01    174  .730  .15578E-01    175  .740  .16117E-01
176  .750  .16650E-01    177  .760  .17175E-01    178  .770  .17691E-01    179  .780  .18196E-01    180  .790  .18686E-01
181  .800  .19160E-01    182  .810  .19613E-01    183  .820  .20041E-01    184  .830  .20442E-01    185  .840  .20808E-01
186  .850  .21134E-01    187  .860  .21414E-01    188  .870  .21638E-01    189  .880  .21796E-01    190  .890  .21876E-01
191  .900  .21862E-01    192  .910  .21734E-01    193  .920  .21469E-01    194  .930  .21032E-01    195  .940  .20379E-01
196  .950  .19447E-01    197  .960  .18140E-01    198  .970  .16316E-01    199  .980  .13647E-01    200  .990  .96614E-02
201 1.000 0.

                                                 LEGENDRE SERIES ORDER =    2
  J    Q          GN          J    Q          GN          J    Q          GN          J    Q          GN          J    Q          GN
  1-1.000 0.                  2 -.990  .42407E-06      3 -.980  .14729E-05      4 -.970  .30000E-05      5 -.960  .49225E-05
  6 -.950  .71801E-05      7 -.940  .97246E-05      8 -.930  .12517E-04      9 -.920  .15521E-04     10 -.910  .18707E-04
 11 -.900  .22049E-04     12 -.890  .25521E-04     13 -.880  .29107E-04     14 -.870  .32771E-04     15 -.860  .36510E-04
 16 -.850  .40300E-04     17 -.840  .44126E-04     18 -.830  .47973E-04     19 -.820  .51827E-04     20 -.810  .55673E-04
 21 -.800  .59501E-04     22 -.790  .63295E-04     23 -.780  .67053E-04     24 -.770  .70756E-04     25 -.760  .74397E-04
 26 -.750  .77967E-04     27 -.740  .81457E-04     28 -.730  .84850E-04     29 -.720  .88163E-04     30 -.710  .91365E-04
 31 -.700  .94456E-04     32 -.690  .97430E-04     33 -.680  .10028E-03     34 -.670  .10300E-03     35 -.660  .10559E-03
 36 -.650  .10803E-03     37 -.640  .11034E-03     38 -.630  .11249E-03     39 -.620  .11448E-03     40 -.610  .11632E-03
 41 -.600  .11800E-03     42 -.590  .11951E-03     43 -.580  .12085E-03     44 -.570  .12203E-03     45 -.560  .12302E-03
 46 -.550  .12384E-03     47 -.540  .12448E-03     48 -.530  .12494E-03     49 -.520  .12521E-03     50 -.510  .12530E-03
 51 -.500  .12520E-03     52 -.490  .12491E-03     53 -.480  .12443E-03     54 -.470  .12376E-03     55 -.460  .12290E-03
 56 -.450  .12185E-03     57 -.440  .12061E-03     58 -.430  .11917E-03     59 -.420  .11754E-03     60 -.410  .11572E-03
 61 -.400  .11371E-03     62 -.390  .11150E-03     63 -.380  .10910E-03     64 -.370  .10652E-03     65 -.360  .10374E-03
 66 -.350  .10078E-03     67 -.340  .97634E-04     68 -.330  .94301E-04     69 -.320  .90785E-04     70 -.310  .87087E-04
 71 -.300  .83211E-04     72 -.290  .79157E-04     73 -.280  .74929E-04     74 -.270  .70529E-04     75 -.260  .65960E-04
 76 -.250  .61224E-04     77 -.240  .56325E-04     78 -.230  .51266E-04     79 -.220  .46050E-04     80 -.210  .40681E-04
 81 -.200  .35162E-04     82 -.190  .29498E-04     83 -.180  .23692E-04     84 -.170  .17750E-04     85 -.160  .11674E-04
 86 -.150  .54704E-05     87 -.140 -.85564E-06     88 -.130 -.73025E-05     89 -.120 -.13862E-04     90 -.110 -.20529E-04
 91 -.100 -.27298E-04     92 -.090 -.34165E-04     93 -.080 -.41122E-04     94 -.070 -.48165E-04     95 -.060 -.55286E-04
 96 -.050 -.62480E-04     97 -.040 -.69740E-04     98 -.030 -.77053E-04     99 -.020 -.84429E-04    100 -.010 -.91845E-04
101  .000 -.99298E-04    102  .010 -.10678E-03    103  .020 -.11429E-03    104  .030 -.12181E-03    105  .040 -.12933E-03
106  .050 -.13586E-03    107  .060 -.14437E-03    108  .070 -.15187E-03    109  .080 -.15934E-03    110  .090 -.16676E-03
111  .100 -.17415E-03    112  .110 -.18147E-03    113  .120 -.18973E-03    114  .130 -.19592E-03    115  .140 -.20302E-03
116  .150 -.21002E-03    117  .160 -.21691E-03    118  .170 -.22369E-03    119  .180 -.23033E-03    120  .190 -.23684E-03
121  .200 -.24319E-03    122  .210 -.24937E-03    123  .220 -.25538E-03    124  .230 -.26120E-03    125  .240 -.26681E-03
126  .250 -.27221E-03    127  .260 -.27738E-03    128  .270 -.28231E-03    129  .280 -.28698E-03    130  .290 -.29138E-03
131  .300 -.29550E-03    132  .310 -.29932E-03    133  .320 -.30283E-03    134  .330 -.30600E-03    135  .340 -.30883E-03
136  .350 -.31130E-03    137  .360 -.31340E-03    138  .370 -.31510E-03    139  .380 -.31640E-03    140  .390 -.31727E-03
141  .400 -.31770E-03    142  .410 -.31766E-03    143  .420 -.31715E-03    144  .430 -.31615E-03    145  .440 -.31464E-03
146  .450 -.31259E-03    147  .460 -.30999E-03    148  .470 -.30683E-03    149  .480 -.30308E-03    150  .490 -.29873E-03
151  .500 -.29375E-03    152  .510 -.28812E-03    153  .520 -.28134E-03    154  .530 -.27467E-03    155  .540 -.26720E-03
156  .550 -.25891E-03    157  .560 -.24967E-03    158  .570 -.23978E-03    159  .580 -.22911E-03    160  .590 -.21763E-03
161  .600 -.20534E-03    162  .610 -.19222E-03    163  .620 -.17824E-03    164  .630 -.16338E-03    165  .640 -.14764E-03
166  .650 -.13100E-03    167  .660 -.11344E-03    168  .670 -.94951E-04    169  .680 -.75523E-04    170  .690 -.55147E-04
171  .700 -.33815E-04    172  .710 -.11523E-04    173  .720  .11731E-04    174  .730  .35947E-04    175  .740  .61120E-04
176  .750  .87238E-04    177  .760  .11428E-03    178  .770  .14222E-03    179  .780  .17102E-03    180  .790  .20064E-03
```

APPENDIX 3

SAW DEVICE INVESTIGATIONS

USER-ORIENTED DOCUMENTATION AND PROGRAM LISTINGS

Program GAL3SE computes the surface charge density expansion coefficients for perfectly conducting, infinitely thin electrodes comprising a planar array impressed with potentials, POL(I). The electrostatic fields thus induced establish an equilibrium distribution of charge on each electrode surface.

The system of Fredhom integral equations of the first kind relating charge density to applied potentials have been reduced to a linear system of algebraic equations using the Galerkin method described in the paper.

"Electric Field Interactions Within Finite Arrays and the Design of Withdrawal Weighted SAW Filters at Fundamental and Higher Harmonics," by Laker, Cohen, and Slobodnik, Jr., 1976. Ultrasonics Symposium Proceedings, IEEE CAT. No. CH1120-5SU, pp. 317-321.

USER must supply the following:

A. in NAMELIST/PARAMS/
   1. ETA = electrode width/center-to-center spacing (=.5)
   2. NP = no. of points used in quadrature to create PSI(X,X,X) = 128 (usually)
   3. NN = no. of terms in charge density expansion (=6)
   4. ITOTAL = max. no. of electrodes in array
   5. NCASE = no. of different cases (potential sequences) to be applied to the array for study.

B. in NAMELIST/VSEQ/
   1. JTH = electrode location for which expansion coeffs. are going to be evaluated.
   2. NUM = outdated parameter, never used.
   3. POL = array of applied potentials.
      NOTE: POL(X) less than EE signifies a withdrawn electrode.

C. Off-diagonal matrix elements on TAPE1 generated from program PSIGEN.

ARRAY DIMENSIONS

B(ITOTAL*NN + 1)
C([ITOTAL*NN+1]*[ITOTAL*NN+1])
PSI(NN,NN, 2*ITOTAL-1)
ISEQ(ITOTAL) -- not currently used in this program
POL(ITOTAL)
CC(NN)
U(ITOTAL)

NOTE: In subroutine COEFFS, the variable JTH identifies the electrode(s) for which the expansion coefficients array is to be evaluated. JTH may be varied in a DO-LOOP or fixed at one value.

Programs GAL3AD (for double electrode arrays) and GAL3AT (for triple electrode arrays) must also be used in the manner described above.

```
      PROGRAM GAL3SE(INPUT,OUTPUT,TAPE1,TAPE6,TAPE7,PUNCH)
      COMMON B(127),C(16129),PSI(6,6,41)
      COMMON PHIQ,ETAKJ,NP,NP2,NP2P1,NUP,ALF
      COMMON JJ,KM,IA(21),ITOTAL,JTH,NN,KASE,ISEQ(21),POL(21),CC(6)
      DIMENSION V(22)
      DIMENSION U(21)
      DATA ZP,ZM,ZA/1H+,1H-,1HA/
      DATA PI/3.14159265358979/
      NAMELIST/PARAMS/ETA,NP,NN,ITOTAL,NCASE
      NAMELIST/VSEQ/NUM,JTH,POL
C
C              VERSION  3A  OF GALERKN  -  2/16/77     SINGLE ELECTRODE ARRAYS
C              COEFF.  ARE COMPUTED FOR  ETA = .5
C
      READ PARAMS
      PRINT PARAMS
C
C
      JTOP=2*ITOTAL-1
      EE=1.E-8
      AL2=ALOG(2.)
      NP2=NP/2
      NP2P1=NP2+1
      NUP=NP2P1+1
      ALF=2.*PI/FLOAT(NP)
      DTDPHI=ALF*ALF
      PISQ=PI*PI
      PILOG=PISQ*AL2
      PIP5=.5*PISQ
      PRINT 400
      ITOP=ITOTAL*NN+1
C
  200 FORMAT(1X,5E20.10)
C
      DO 794 I=1,486
  794 READ(7,VSEQ)
      READ(1,PARAMS)
      PRINT PARAMS
      READ(1,200) (((PSI(L1,L2,L3),L1=1,NN),L2=1,NN),L3=1,JTOP)
  333 CONTINUE
      DO 999 KASE=1,243
      READ(7,VSEQ)
      J=0
      DO 27 I=1,ITOTAL
      IA(I)=1
      AA=ABS(POL(I))
      IF(AA.GT.EE)J=J+1
      IF(AA.LE.EE)IA(I)=0
   27 CONTINUE
C
      IDIM=J*NN+1
      MOST=IDIM*IDIM
      DO 272 I=1,MOST
  272 C(I)=0.
      K=0
      DO 277 I=1,ITOP
  277 B(I)=0.
      DO 160 KKK=1,ITOTAL
      IF(IA(KKK).EQ.0)GO TO 160
      K=K+1
      U(K)=POL(KKK)
      I=1+(K-1)*NN
  160 CONTINUE
C
C
C         FORMATION OF THE OFF-DIAGONAL ELEMENTS  A(IROW,ICOL)
C
      DO 250 L=1,NN
      DO 350 N=1,NN
      JJ=0
      DO 450 JJJ=1,ITOTAL
      IF(IA(JJJ).EQ.0)GO TO 450
      JJ=JJ+1
      K=0
      DO 550 KKK=1,ITOTAL
      IF(IA(KKK).EQ.0)GO TO 550
      K=K+1
      IF(K.EQ.JJ)GO TO 550
      KMJP7=KKK-JJJ+ITOTAL
      IROW=L+(K-1)*NN
      ICOL=N+(JJ-1)*NN
      I=IROW+(ICOL-1)*IDIM
      C(I)=PSI(L,N,KMJP7)
  550 CONTINUE
  450 CONTINUE
  350 CONTINUE
  250 CONTINUE
C
C         COMPUTATION OF DIAGONAL MATRIX ELEMENTS
C         L=N=0  ELEMENTS.    K=JJ=1,2,...,J
C
      NI=NN*(IDIM+1)
      I=1-NI
      DO 67 K=1,J
      I=I+NI
      C(I)=-PILOG
   67 CONTINUE
C
C     L=N=1,2,...,NN  MATRIX ELEMENTS       K=JJ=1,2,...,J
C
      DO 77 L=2,NN
      DO 87 K=1,J
      II=L+(K-1)*NN
      I=II+(II-1)*IDIM
      C(I)=-PIP5/FLOAT(L-1)
   87 CONTINUE
   77 CONTINUE
C
C
C     MATRIX ELEMENTS FOR ZERO TOTAL CHARGE CONDITION & REF. POTENTIAL
C
      DO 99 K=1,J
      IR=1+(K-1)*NN
      I=IR+(IDIM-1)*IDIM
      C(I)=PI
      DO 89 L=2,NN
      IROW=L+(K-1)*NN
      II=IROW+(IDIM-1)*IDIM
      C(II)=0.
   89 CONTINUE
   99 CONTINUE
C
      DO 79 K=1,J
      IC=1+(K-1)*NN
      I=IDIM+(IC-1)*IDIM
      C(I)=1.
   79 CONTINUE
```

```
C
C
      CALL SIMQ(C, B, IDIM, KS)
C
C
C
C
C
      CALL COEFFS
C
  400 FORMAT(1H1, 30X, *CHARGE DENSITY EXPANSION COEFFICIENTS FOR  ETA=.5
     1     -      NOT NORMALIZED TO SMITHS TABLE*)
  290 FORMAT(1X, 2I2, 1X, 6E11.4)
  291 FORMAT(1X, 2I2, 1X, 17(A1, 2X))
  349 FORMAT(3X, 17(A1, 2X), 24X, I2)
  351 FORMAT(5X, 5F13.10)
  149 FORMAT(/45X, 17(A1, 2X)/)
  151 FORMAT(30X, 5F13.10)
  999 CONTINUE
C
C
      END
      SUBROUTINE COEFFS
      COMMON B(127), C(16129), PSI(6, 6, 41)
      COMMON DUM(7)
      COMMON JJ, KM, IA(21), ITOTAL, JTH, NN, KASE, ISEQ(21), POL(21), CC(6)
      NAMELIST/CVAR/KASE, JTH, KM, MM, POL, CC
C
      JTH=11
      JJ=0  $  KM=0
      DO 199 KA=1, ITOTAL
      II=IA(KA)
      JJ=JJ+1
      IF(II.EQ.0)GO TO 199
      KM=KM+1
      IF(JJ.NE.JTH)GO TO 199
      MM=(KM-1)*NN
C
C
      DO 200 N=1, NN
  200 CC(N)=B(N+MM)
      WRITE(6, CVAR)
      PUNCH 45, KASE, JJ, (POL(MA), MA=1, ITOTAL)
      PRINT 45, KASE, JJ, (POL(MA), MA=1, ITOTAL)
      PUNCH 55, KASE, JJ, CC
      PRINT 55, KASE, JJ, CC
   45 FORMAT(1X, I3, 1X, I2, 3X, 7F10.6/(10X, 7F10.6))
   55 FORMAT(1X, I3, 1X, I2, 3X, 7F10.6)
  199 CONTINUE
C
      RETURN
      END
      SUBROUTINE SIMQ(A, B, N, KS)                                       -F
      DIMENSION A(1), B(1)                                               -F
C                                                                        -F
C        SUBROUTINE SIMQ                                                 -F
C                                                                        -F
C        PURPOSE                                                         -F
C           OBTAIN SOLUTION OF A SET OF SIMULTANEOUS LINEAR EQUATIONS,   -F
C           AX=B                                                         -F
C                                                                        -F
C        USAGE                                                           -F
C           CALL SIMQ(A, B, N, KS)                                       -F
C                                                                        -F
```

```
C          A - MATRIX OF COEFFICIENTS STORED COLUMNWISE.  THESE ARE      -F
C              DESTROYED IN THE COMPUTATION.  THE SIZE OF MATRIX A IS    -F
C              N BY N                                                    -F
C          B - VECTOR OF ORIGINAL CONSTANTS (LENGTH N). THESE ARE        -F
C              REPLACED BY FINAL SOLUTION VALUES, VECTOR X.              -F
C          N - NUMBER OF EQUATIONS AND VARIABLES. N MUST BE .GT. ONE.    -F
C          KS - OUTPUT DIGIT                                             -F
C                0 FOR A NORMAL SOLUTION                                 -F
C                1 FOR A SINGULAR SET OF EQUATIONS                       -F
C                                                                        -F
C       REMARKS                                                          -F
C          MATRIX A MUST BE GENERAL.                                     -F
C          IF MATRIX IS SINGULAR , SOLUTION VALUES ARE MEANINGLESS.      -F
C          AN ALTERNATIVE SOLUTION MAY BE OBTAINED BY USING MATRIX       -F
C          INVERSION (MINV) AND MATRIX PRODUCT (GMPRD).                  -F
C                                                                        -F
C       SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                    -F
C          NONE                                                          -F
C                                                                        -F
C       METHOD                                                           -F
C          METHOD OF SOLUTION IS BY ELIMINATION USING LARGEST PIVOTAL    -F
C          DIVISOR. EACH STAGE OF ELIMINATION CONSISTS OF INTERCHANGING  -F
C          ROWS WHEN NECESSARY TO AVOID DIVISION BY ZERO OR SMALL        -F
C          ELEMENTS.                                                     -F
C          THE FORWARD SOLUTION TO OBTAIN VARIABLE N IS DONE IN          -F
C          N STAGES. THE BACK SOLUTION FOR THE OTHER VARIABLES IS        -F
C          CALCULATED BY SUCCESSIVE SUBSTITUTIONS. FINAL SOLUTION        -F
C          VALUES ARE DEVELOPED IN VECTOR B, WITH VARIABLE 1 IN B(1),    -F
C          VARIABLE 2 IN B(2),.........., VARIABLE N IN B(N).            -F
C          IF NO PIVOT CAN BE FOUND EXCEEDING A TOLERANCE OF 0.0,        -F
C          THE MATRIX IS CONSIDERED SINGULAR AND KS IS SET TO 1. THIS    -F
C          TOLERANCE CAN BE MODIFIED BY REPLACING THE FIRST STATEMENT.   -F
C                                                                        -F
C                                                                        -F
C       FORWARD SOLUTION                                                 -F
C                                                                        -F
      TOL=0.0                                                            -F
      KS=0                                                               -F
      JJ=-N                                                              -F
      DO 65 J=1,N                                                        -F
      JY=J+1                                                             -F
      JJ=JJ+N+1                                                          -F
      BIGA=0                                                             -F
      IT=JJ-J                                                            -F
      DO 30 I=J,N                                                        -F
C                                                                        -F
C       SEARCH FOR MAXIMUM COEFFICIENT IN COLUMN                         -F
C                                                                        -F
      IJ=IT+I                                                            -F
      IF(ABS(BIGA)-ABS(A(IJ))) 20,30,30                                  -F
   20 BIGA=A(IJ)                                                         -F
      IMAX=I                                                             -F
   30 CONTINUE                                                           -F
C                                                                        -F
C       TEST FOR PIVOT LESS THAN TOLERANCE (SINGULAR MATRIX)             -F
C                                                                        -F
      IF(ABS(BIGA)-TOL) 35,35,40                                         -F
   35 KS=1                                                               -F
      RETURN                                                             -F
C                                                                        -F
C       INTERCHANGE ROWS IF NECESSARY                                    -F
C                                                                        -F
   40 I1=J+N*(J-2)                                                       -F
      IT=IMAX-J                                                          -F
      DO 50 K=J,N                                                        -F
      I1=I1+N                                                           -F
      I2=I1+IT                                                          -F
      SAVE=A(I1)                                                        -F
      A(I1)=A(I2)                                                       -F
      A(I2)=SAVE                                                        -F
                                                                        -F
C                                                                       -F
C        DIVIDE EQUATION BY LEADING COEFFICIENT                         -F
C                                                                       -F
   50 A(I1)=A(I1)/BIGA                                                  -F
      SAVE=B(IMAX)                                                      -F
      B(IMAX)=B(J)                                                      -F
      B(J)=SAVE/BIGA                                                    -F
C                                                                       -F
C        ELIMINATE NEXT VARIABLE                                        -F
C                                                                       -F
      IF(J-N) 55,70,55                                                  -F
   55 IQS=N*(J-1)                                                       -F
      DO 65 IX=JY,N                                                     -F
      IXJ=IQS+IX                                                        -F
      IT=J-IX                                                           -F
      DO 60 JX=JY,N                                                     -F
      IXJX=N*(JX-1)+IX                                                  -F
      JJX=IXJX+IT                                                       -F
   60 A(IXJX)=A(IXJX)-(A(IXJ)*A(JJX))                                   -F
   65 B(IX)=B(IX)-(B(J)*A(IXJ))                                         -F
C                                                                       -F
C        BACK SOLUTION                                                  -F
C                                                                       -F
   70 NY=N-1                                                            -F
      IT=N*N                                                            -F
      DO 80 J=1,NY                                                      -F
      IA=IT-J                                                           -F
      IB=N-J                                                            -F
      IC=N                                                              -F
      DO 80 K=1,J                                                       -F
      B(IB)=B(IB)-A(IA)*B(IC)                                           -F
      IA=IA-N                                                           -F
   80 IC=IC-1                                                           -F
      RETURN                                                            -F
      END                                                               -F
\
      PROGRAM GAL3AD(INPUT,OUTPUT,TAPE1,TAPE6,TAPE7)
      COMMON B(133),C(17689)
      COMMON PSI(6,6,43)
      COMMON PHIQ,ETAKJ,NP,NP2,NP2P1,NUP,ALF
      COMMON JJ,KM,IA(22),ITOTAL,KTH,NN,KASE,SEQ(22),V(22),CC(10)
      DIMENSION U(22)
      DATA ZP,ZM,ZA/1H+,1H-,1HA/
      DATA PI/3.14159265358979/
      NAMELIST/PARAMS/ETA,NP,NN,ITOTAL,NCASE
      NAMELIST/VSEQ/V
C
C ------------          VERSION 3AD OF GALERKN  -. 3/14/77  -----
C ------------          DOUBLE ELECTRODES.
C ------------          DOUBLE ELECTRODE  COEFF  ARE COMPUTED FOR  ETA =  5
C
      READ PARAMS
      PRINT PARAMS
C
C ----    BEGIN THIS RUN AT CASE #  198
      DO 727 I=1,195
 727      READ(7,VSEQ)
C
      JTOP=2*ITOTAL-1
      AL2=ALOG(2.)
      NP2=NP/2
      NP2P1=NP2+1
      NUP=NP2P1+1
      ALF=2.*PI/FLOAT(NP)
      DTDPHI=ALF*ALF
      PISQ=PI*PI
      PILOG=PISQ*AL2
      PIP5=.5*PISQ
      PRINT 400
      ITOP=ITOTAL*NN+1
C
 200  FORMAT(1X,5E20.10)
C
      READ(1,PARAMS)
      PRINT PARAMS
      READ(1,200) (((PSI(L1,L2,L3),L1=1,NN),L2=1,NN),L3=1,JTOP)
 333  CONTINUE
      DO 999 KASE=1,NCASE
      READ(7,VSEQ)
      J=0
      DO 27 I=1,ITOTAL
      IA(I)=1
      INTV=IFIX(V(I))
      SEQ(I)=ZA
      IF(INTV.EQ.1)SEQ(I)=ZP
      IF(INTV.EQ.-1)SEQ(I)=ZM
      IF(INTV.EQ.0)IA(I)=0
      IF(INTV.EQ.0)GO TO 27
      J=J+1
  27  CONTINUE
C
      IDIM=J*NN+1
      MOST=IDIM*IDIM
      DO 272 I=1,MOST
 272  C(I)=0.
      K=0
      DO 277 I=1,ITOP
 277  B(I)=0
```

```
      DO 160 KKK=1,ITOTAL
      IF(IA(KKK).EQ.0)GO TO 160
      K=K+1
      U(K)=V(KKK)
      I=1+(K-1)*NN
      B(I)=PI*U(K)
  160 CONTINUE
C
C
C         FORMATION OF THE OFF-DIAGONAL ELEMENTS  A(IROW,ICOL)
C
      DO 250 L=1,NN
      DO 350 N=1,NN
      JJ=0
      DO 450 JJJ=1,ITOTAL
      IF(IA(JJJ).EQ.0)GO TO 450
      JJ=JJ+1
      K=0
      DO 550 KKK=1,ITOTAL
      IF(IA(KKK).EQ.0)GO TO 550
      K=K+1
      IF(K.EQ.JJ)GO TO 550
      KMJP7=KKK-JJJ+ITOTAL
      IROW=L+(K-1)*NN
      ICOL=N+(JJ-1)*NN
      I=IROW+(ICOL-1)*IDIM
      C(I)=PSI(L,N,KMJP7)
  550 CONTINUE
  450 CONTINUE
  350 CONTINUE
  250 CONTINUE
C
C         COMPUTATION OF DIAGONAL MATRIX ELEMENTS
C         L=N=0  ELEMENTS.    K=JJ=1,2,...,J
C
      NI=NN*(IDIM+1)
      I=1-NI
      DO 67 K=1,J
      I=I+NI
      C(I)=-PILOG
   67 CONTINUE
C
C     L=N=1,2,...,NN  MATRIX ELEMENTS.       K=JJ=1,2,...,J
C
      DO 77 L=2,NN
      DO 87 K=1,J
      II=L+(K-1)*NN
      I=II+(II-1)*IDIM
      C(I)=-PIP5/FLOAT(L-1)
   87 CONTINUE
   77 CONTINUE
C
C
C     MATRIX ELEMENTS FOR ZERO TOTAL CHARGE CONDITION & REF. POTENTIAL
C
      DO 99 K=1,J
      IR=1+(K-1)*NN
      I=IR+(IDIM-1)*IDIM
      C(I)=PI
      DO 89 L=2,NN
      IROW=L+(K-1)*NN
      II=IROW+(IDIM-1)*IDIM
      C(II)=0.
   89 CONTINUE
```

```
      IC=1+(K-1)*NN
      I=IDIM+(IC-1)*IDIM
      C(I)=1.
   79 CONTINUE
C
C
C
C
      CALL SIMQ(C,B,IDIM,KS)
      DO 24 I=1,IDIM
   24 C(I)=B(I)
C
C
C
C
C
      CALL COEFFSD
C
  400 FORMAT(1H1,30X,*CHARGE DENSITY EXPANSION COEFFICIENTS FOR  ETA=.5
     1  -     NOT NORMALIZED TO SMITHS TABLE*)
  290 FORMAT(1X,2I2,1X,6E11.4)
  291 FORMAT(1X,2I2,1X,17(A1,2X))
  349 FORMAT(3X,17(A1,2X),24X,I2)
  351 FORMAT(5X,5F13.10)
  149 FORMAT(/45X,17(A1,2X)/)
  151 FORMAT(30X,5F13.10)
  999 CONTINUE
C
C
      END
      SUBROUTINE COEFFSD
      COMMON B(133),C(17689)
      COMMON PSI(6,6,43)
      COMMON DUM(7)
      COMMON JJ,KM,IA(22),ITOTAL,KTH,NN,KASE,SEQ(22),V(22),CC(10)
      DIMENSION IK(22)
      NAMELIST/CVAR/KASE,JJ,KM,MM,V,CC
C
C
C    *********************FOR DOUBLE ELECTRODE ARRAYS *************
C          KR & KL ARE FOR THE MIDDLE ELECTRODES
           KR=11
           KL=12
C    *********************FOR DOUBLE ELECTRODE ARRAYS *************
      JJ=0  $  KM=0
      DO 199 KA=1,ITOTAL
      II=IA(KA)
      JJ=JJ+1
      IF(II.EQ.0)GO TO 199
      KM=KM+1
      IF(JJ.NE.KL.AND.JJ.NE.KR)GO TO 199
      MM=(KM-1)*NN
C
      PRINT 115,KASE,JJ,KM,MM
      PRINT 125,(SEQ(I),I=1,ITOTAL)
      PRINT 135,(C(N+MM),N=1,NN)
C
      DO 200 N=1,NN
  200 CC(N)=C(N+MM)
      WRITE(6,CVAR)
  199 CONTINUE
C
  115 FORMAT(///15X,*CASE*,I4,5X,*LOCATION NO.*,I3,5X,*KM=*,I3,5X,
     1  *MM=*,I3)
  125 FORMAT(5X,5F13.10)
  125 FORMAT(/10X,22(A1,2X)/)
      RETURN
      END
      SUBROUTINE SIMQ(A,B,N,KS)                                         -F
      DIMENSION A(1),B(1)                                               -F
C                                                                       -F
C        SUBROUTINE SIMQ                                                -F
C                                                                       -F
C        PURPOSE                                                        -F
C           OBTAIN SOLUTION OF A SET OF SIMULTANEOUS LINEAR EQUATIONS,  -F
C           AX=B                                                        -F
C                                                                       -F
C        USAGE                                                          -F
C           CALL SIMQ(A,B,N,KS)                                         -F
C                                                                       -F
C        DESCRIPTION OF PARAMETERS                                      -F
C           A - MATRIX OF COEFFICIENTS STORED COLUMNWISE.  THESE ARE    -F
C               DESTROYED IN THE COMPUTATION.  THE SIZE OF MATRIX A IS  -F
C               N BY N.                                                 -F
C           B - VECTOR OF ORIGINAL CONSTANTS (LENGTH N). THESE ARE      -F
C               REPLACED BY FINAL SOLUTION VALUES, VECTOR X.            -F
C           N - NUMBER OF EQUATIONS AND VARIABLES. N MUST BE .GT. ONE.  -F
C           KS - OUTPUT DIGIT                                           -F
C                0 FOR A NORMAL SOLUTION                                -F
C                1 FOR A SINGULAR SET OF EQUATIONS                      -F
C                                                                       -F
C        REMARKS                                                        -F
C           MATRIX A MUST BE GENERAL.                                   -F
C           IF MATRIX IS SINGULAR , SOLUTION VALUES ARE MEANINGLESS.    -F
C           AN ALTERNATIVE SOLUTION MAY BE OBTAINED BY USING MATRIX     -F
C           INVERSION (MINV) AND MATRIX PRODUCT (GMPRD).                -F
C                                                                       -F
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                  -F
C           NONE                                                        -F
C                                                                       -F
C        METHOD                                                         -F
C           METHOD OF SOLUTION IS BY ELIMINATION USING LARGEST PIVOTAL  -F
C           DIVISOR. EACH STAGE OF ELIMINATION CONSISTS OF INTERCHANGING -F
C           ROWS WHEN NECESSARY TO AVOID DIVISION BY ZERO OR SMALL      -F
C           ELEMENTS.                                                   -F
C           THE FORWARD SOLUTION TO OBTAIN VARIABLE N IS DONE IN        -F
C           N STAGES. THE BACK SOLUTION FOR THE OTHER VARIABLES IS      -F
C           CALCULATED BY SUCCESSIVE SUBSTITUTIONS. FINAL SOLUTION      -F
C           VALUES ARE DEVELOPED IN VECTOR B, WITH VARIABLE 1 IN B(1),  -F
C           VARIABLE 2 IN B(2),........, VARIABLE N IN B(N).            -F
C           IF NO PIVOT CAN BE FOUND EXCEEDING A TOLERANCE OF 0.0,      -F
C           THE MATRIX IS CONSIDERED SINGULAR AND KS IS SET TO 1. THIS  -F
C           TOLERANCE CAN BE MODIFIED BY REPLACING THE FIRST STATEMENT. -F
C                                                                       -F
C                                                                       -F
C        FORWARD SOLUTION                                               -F
C                                                                       -F
      TOL=0.0                                                           -F
      KS=0                                                              -F
      JJ=-N                                                             -F
      DO 65 J=1,N                                                       -F
      JY=J+1                                                            -F
      JJ=JJ+N+1                                                         -F
      BIGA=0                                                            -F
      IT=JJ-J                                                           -F
      DO 30 I=J,N                                                       -F
C                                                                       -F
C        SEARCH FOR MAXIMUM COEFFICIENT IN COLUMN                       -F
C                                                                       -F
```

```
      IMAX=I                                                       -F
   30 CONTINUE                                                     -F
C                                                                  -F
C        TEST FOR PIVOT LESS THAN TOLERANCE (SINGULAR MATRIX)      -F
C                                                                  -F
      IF(ABS(BIGA)-TOL) 35,35,40                                   -F
   35 KS=1                                                         -F
      RETURN                                                       -F
C                                                                  -F
C        INTERCHANGE ROWS IF NECESSARY                             -F
C                                                                  -F
   40 I1=J+N*(J-2)                                                 -F
      IT=IMAX-J                                                    -F
      DO 50 K=J,N                                                  -F
      I1=I1+N                                                      -F
      I2=I1+IT                                                     -F
      SAVE=A(I1)                                                   -F
      A(I1)=A(I2)                                                  -F
      A(I2)=SAVE                                                   -F
                                                                   -F
C                                                                  -F
C        DIVIDE EQUATION BY LEADING COEFFICIENT                    -F
C                                                                  -F
   50 A(I1)=A(I1)/BIGA                                             -F
      SAVE=B(IMAX)                                                 -F
      B(IMAX)=B(J)                                                 -F
      B(J)=SAVE/BIGA                                               -F
C                                                                  -F
C        ELIMINATE NEXT VARIABLE                                   -F
C                                                                  -F
      IF(J-N) 55,70,55                                             -F
   55 IQS=N*(J-1)                                                  -F
      DO 65 IX=JY.N                                                -F
      IXJ=IQS+IX                                                   -F
      IT=J-IX                                                      -F
      DO 60 JX=JY,N                                                -F
      IXJX=N*(JX-1)+IX                                             -F
      JJX=IXJX+IT                                                  -F
   60 A(IXJX)=A(IXJX)-(A(IXJ)*A(JJX))                              -F
   65 B.IX)=B(IX)-(B(J)*A(IXJ))                                    -F
C                                                                  -F
C        BACK SOLUTION                                             -F
C                                                                  -F
   70 NY=N-1                                                       -F
      IT=N*N                                                       -F
      DO 80 J=1,NY                                                 -F
      IA=IT-J                                                      -F
      IB=N-J                                                       -F
      IC=N                                                         -F
      DO 80 K=1,J                                                  -F
      B(IB)=B(IB)-A(IA)*B(IC)                                      -F
      IA=IA-N                                                      -F
   80 IC=IC-1                                                      -F
      RETURN                                                       -F
      END                                                          -F
00
\
      PROGRAM GAL3AT(INPUT, OUTPUT, TAPE1, TAPE6, TAPE7, PUNCH)
      COMMON B(235), C(55225)
      COMMON PSI(6, 6, 77)
      COMMON PHIQ, ETAKJ, NP, NP2, NP2P1, NUP, ALF
      COMMON JJ, KM, IA(39), ITOTAL, JTH, NN, KASE, ISEQ(13), POL(39), CC(6)
      DIMENSION U(39)
      INTEGER POL
      DATA ZP, ZM, ZA/1H+, 1H-, 1HA/
      DATA PI/3.14159265358979/
      NAMELIST/PARAMS/ETA, NP, NN, ITOTAL, NCASE
      NAMELIST/VSEQ/JTH, NE, ISEQ, POL
C
C               VERSION  3A  OF GALERKN   -  2/16/77
C               COEFF.  ARE COMPUTED FOR  ETA = .5
C
C               TRIPLET COMPUTATIONS
                JTH=20
C
C
      READ PARAMS
      PRINT PARAMS
C
C
      JTOP=2*ITOTAL-1
      AL2=ALOG(2.)
      NP2=NP/2
      NP2P1=NP2+1
      NUP=NP2P1+1
      ALF=2.*PI/FLOAT(NP)
      DTDPHI=ALF*ALF
      PISQ=PI*PI
      PILOG=PISQ*AL2
      PIP5=.5*PISQ
      PRINT 400
      ITOP=ITOTAL*NN+1
C
  200 FORMAT(1X, 5E20.10)
C
      READ(1, PARAMS)
      PRINT PARAMS
      READ(1, 200) (((PSI(L1, L2, L3), L1=1, NN), L2=1, NN), L3=1, JTOP)
C
      DO 999 KASE=1, NCASE
      READ(7, VSEQ)
      PRINT VSEQ
      J=0
      DO 27 I=1, ITOTAL
      IA(I)=1
      INTV=POL(I)
      IF(INTV.EQ.0)IA(I)=0
      IF(INTV.EQ.0)GO TO 27
      J=J+1
   27 CONTINUE
C
      IDIM=J*NN+1
      MOST=IDIM*IDIM
      DO 272 I=1, MOST
  272 C(I)=0.
      K=0
      DO 277 I=1, ITOP
      K=K+1
      U(K)=POL(KKK)
      I=1+(K-1)*NN
      B(I)=PI*U(K)
  160 CONTINUE
C
C
C         FORMATION OF THE OFF-DIAGONAL ELEMENTS  A(IROW,ICOL)
C
      DO 250 L=1,NN
      DO 350 N=1,NN
      JJ=0
      DO 450 JJJ=1,ITOTAL
      IF(IA(JJJ).EQ.0)GO TO 450
      JJ=JJ+1
      K=0
      DO 550 KKK=1,ITOTAL
      IF(IA(KKK).EQ.0)GO TO 550
      K=K+1
      IF(K.EQ.JJ)GO TO 550
      KMJP7=KKK-JJJ+ITOTAL
      IROW=L+(K-1)*NN
      ICOL=N+(JJ-1)*NN
      I=IROW+(ICOL-1)*IDIM
      C(I)=PSI(L,N,KMJP7)
  550 CONTINUE
  450 CONTINUE
  350 CONTINUE
  250 CONTINUE
C
C         COMPUTATION OF DIAGONAL MATRIX ELEMENTS
C         L=N=0  ELEMENTS.   K=JJ=1,2,...,J
C
      NI=NN*(IDIM+1)
      I=1-NI
      DO 67 K=1,J
      I=I+NI
      C(I)=-PILOG
   67 CONTINUE
C
C     L=N=1,2,...,NN  MATRIX ELEMENTS      K=JJ=1,2,...,J
C
      DO 77 L=2,NN
      DO 87 K=1,J
      II=L+(K-1)*NN
      I=II+(II-1)*IDIM
      C(I)=-PIP5/FLOAT(L-1)
   87 CONTINUE
   77 CONTINUE
C
C
C     MATRIX ELEMENTS FOR ZERO TOTAL CHARGE CONDITION & REF. POTENTIAL
C
      DO 99 K=1,J
      IR=1+(K-1)*NN
      I=IR+(IDIM-1)*IDIM
      C(I)=PI
      DO 89 L=2,NN
      IROW=L+(K-1)*NN
      II=IROW+(IDIM-1)*IDIM
      C(II)=0.
   89 CONTINUE
   99 CONTINUE
C
      DO 79 K=1,J
      IC=1+(K-1)*NN
      I=IDIM+(IC-1)*IDIM
      C(I)=1.
   79 CONTINUE
C
C
C
C
      CALL SIMQ(C,B,IDIM,KS)
      CALL COEFFT
C
  400 FORMAT(1H1,30X,*CHARGE DENSITY EXPANSION COEFFICIENTS FOR  ETA=.5
     1  -     NOT NORMALIZED TO SMITHS TABLE*)
  999 CONTINUE
C
C
      END
      SUBROUTINE COEFFT
      COMMON B(235),C(55225)
      COMMON PSI(6,6,77)
      COMMON DUM(7)
      COMMON JD,KM,IA(39),ITOTAL,JTH,NN,KASE,ISEQ(13),POL(39),CC(6)
      INTEGER POL
      NAMELIST/CVAR/KASE,JJ,KM,MM,POL,CC
C -----   COEFF. FOR 3-PHASE TRIPLETS  -----------
      JP=JTH+1
      JM=JTH-1
      JJ=KM=0
      DO 199 KA=1,ITOTAL
      II=IA(KA)
      JJ=JJ+1
      IF(II.EQ.0)GO TO 199
      KM=KM+1
      IF(JJ.LT.JM.OR.JJ.GT.JP)GO TO 199
      MM=(KM-1)*NN
C
      IF(JM.EQ.JJ)PRINT 225,ISEQ,(B(N+MM),N=1,NN)
      IF(JTH.EQ.JJ)PRINT 235,(B(N+MM),N=1,NN)
      IF(JP.EQ.JJ)PRINT 245,(B(N+MM),N=1,NN)
C
      DO 200 N=1,NN
  200 CC(N)=B(N+MM)
C
      WRITE(6,CVAR)
  199 CONTINUE
C
  225 FORMAT(30X,*CONFIGURATION OF 3-PHASE TRIPLETS*//30X,13I2///5X,
     1 *LEFT*,10X,6F9.6)
  235 FORMAT(5X,*CENTER*,8X,6F9.6)
  245 FORMAT(5X,*RIGHT*,9X,6F9.6///)
      RETURN
      END
      SUBROUTINE SIMQ(A,B,N,KS)                                        -F
      DIMENSION A(1),B(1)                                              -F
C                                                                      -F
C        SUBROUTINE SIMQ                                               -F
C                                                                      -F
C        PURPOSE                                                       -F
C           OBTAIN SOLUTION OF A SET OF SIMULTANEOUS LINEAR EQUATIONS, -F
C           AX=B                                                       -F
C                                                                      -F
C        USAGE                                                         -F
C           CALL SIMQ(A,B,N,KS)                                        -F
C                                                                      -F
```

```
C               N BY N                                                   -F
C           B - VECTOR OF ORIGINAL CONSTANTS (LENGTH N).  THESE ARE      -F
C               REPLACED BY FINAL SOLUTION VALUES, VECTOR X.             -F
C           N - NUMBER OF EQUATIONS AND VARIABLES.  N MUST BE .GT. ONE.  -F
C           KS - OUTPUT DIGIT                                            -F
C                0 FOR A NORMAL SOLUTION                                 -F
C                1 FOR A SINGULAR SET OF EQUATIONS                       -F
C                                                                        -F
C        REMARKS                                                         -F
C           MATRIX A MUST BE GENERAL.                                    -F
C           IF MATRIX IS SINGULAR , SOLUTION VALUES ARE MEANINGLESS.     -F
C           AN ALTERNATIVE SOLUTION MAY BE OBTAINED BY USING MATRIX      -F
C           INVERSION (MINV) AND MATRIX PRODUCT (GMPRD)                  -F
C                                                                        -F
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                   -F
C           NONE                                                         -F
C                                                                        -F
C        METHOD                                                          -F
C           METHOD OF SOLUTION IS BY ELIMINATION USING LARGEST PIVOTAL   -F
C           DIVISOR. EACH STAGE OF ELIMINATION CONSISTS OF INTERCHANGING -F
C           ROWS WHEN NECESSARY TO AVOID DIVISION BY ZERO OR SMALL       -F
C           ELEMENTS.                                                    -F
C           THE FORWARD SOLUTION TO OBTAIN VARIABLE N IS DONE IN         -F
C           N STAGES.  THE BACK SOLUTION FOR THE OTHER VARIABLES IS      -F
C           CALCULATED BY SUCCESSIVE SUBSTITUTIONS.  FINAL SOLUTION      -F
C           VALUES ARE DEVELOPED IN VECTOR B, WITH VARIABLE 1 IN B(1),   -F
C           VARIABLE 2 IN B(2),........, VARIABLE N IN B(N).             -F
C           IF NO PIVOT CAN BE FOUND EXCEEDING A TOLERANCE OF 0.0,       -F
C           THE MATRIX IS CONSIDERED SINGULAR AND KS IS SET TO 1.  THIS  -F
C           TOLERANCE CAN BE MODIFIED BY REPLACING THE FIRST STATEMENT.  -F
C                                                                        -F
C                                                                        -F
C        FORWARD SOLUTION                                                -F
C                                                                        -F
      TOL=0.0                                                            -F
      KS=0                                                               -F
      JJ=-N                                                              -F
      DO 65 J=1,N                                                        -F
      JY=J+1                                                             -F
      JJ=JJ+N+1                                                          -F
      BIGA=0                                                             -F
      IT=JJ-J                                                            -F
      DO 30 I=J,N                                                        -F
C                                                                        -F
C        SEARCH FOR MAXIMUM COEFFICIENT IN COLUMN                        -F
C                                                                        -F
      IJ=IT+I                                                            -F
      IF(ABS(BIGA)-ABS(A(IJ))) 20,30,30                                  -F
   20 BIGA=A(IJ)                                                         -F
      IMAX=I                                                             -F
   30 CONTINUE                                                           -F
C                                                                        -F
C        TEST FOR PIVOT LESS THAN TOLERANCE (SINGULAR MATRIX)            -F
C                                                                        -F
      IF(ABS(BIGA)-TOL) 35,35,40                                         -F
   35 KS=1                                                               -F
      RETURN                                                             -F
C                                                                        -F
C        INTERCHANGE ROWS IF NECESSARY                                   -F
C                                                                        -F
   40 I1=J+N*(J-2)                                                       -F
      IT=IMAX-J                                                          -F
      DO 50 K=J,N                                                        -F
      I1=I1+N                                                            -F
      I2=I1+IT                                                           -F
      SAVE=A(I1)                                                        -F
      A(I1)=A(I2)                                                       -F
      A(I2)=SAVE                                                        -F
                                                                        -F
C                                                                       -F
C        DIVIDE EQUATION BY LEADING COEFFICIENT                         -F
C                                                                       -F
   50 A(I1)=A(I1)/BIGA                                                  -F
      SAVE=B(IMAX)                                                      -F
      B(IMAX)=B(J)                                                      -F
      B(J)=SAVE/BIGA                                                    -F
C                                                                       -F
C        ELIMINATE NEXT VARIABLE                                        -F
C                                                                       -F
      IF(J-N) 55,70,55                                                  -F
   55 IQS=N*(J-1)                                                       -F
      DO 65 IX=JY,N                                                     -F
      IXJ=IQS+IX                                                        -F
      IT=J-IX                                                           -F
      DO 60 JX=JY,N                                                     -F
      IXJX=N*(JX-1)+IX                                                  -F
      JJX=IXJX+IT                                                       -F
   60 A(IXJX)=A(IXJX)-(A(IXJ)*A(JJX))                                   -F
   65 B(IX)=B(IX)-(B(J)*A(IXJ))                                         -F
C                                                                       -F
C        BACK SOLUTION                                                  -F
C                                                                       -F
   70 NY=N-1                                                            -F
      IT=N*N                                                            -F
      DO 80 J=1,NY                                                      -F
      IA=IT-J                                                           -F
      IB=N-J                                                            -F
      IC=N                                                              -F
      DO 80 K=1,J                                                       -F
      B(IB)=B(IB)-A(IA)*B(IC)                                           -F
      IA=IA-N                                                           -F
   80 IC=IC-1                                                           -F
      RETURN                                                            -F
      END                                                               -F
00

\
\\ \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\
      PROGRAM PSIGEN(INPUT,OUTPUT,TAPE1)                                XX       2
      COMMON Y(128),YN(6,128),PSI(6,6,77)                               XX       1
      COMMON PHIO,ETAKJ,NP,NP2,NP2P1,NUP,ALF                            CB       2
      COMPLEX Y                                                         MAIN     5
      NAMELIST/PARAMS/ETA,NP,NN,ITOTAL                                  MAIN     6
C                                                                       FIN      1
C                                                                       FIN      2
C --- THIS PROGRAM COMPUTES THE OFF-DIAGONAL MATRIX ELEMENTS FOR ALL    FIN      3
C --- GALERKIN PROGRAMS WRITTEN PRIOR TO  JULY 15, 1978.                FIN      4
C --- THE OFF-DIAGONAL MATRIX ELEMENTS CAN BE WRITTEN AS DOUBLE FINITE  FIN      5
C --- FOURIER TRANSFORMS ONCE THE GALERKIN METHOD IS APPLIED TO EQ.5 IN FIN      6
C --- THE PAPER                                                         FIN      7
C                                                                       FIN      8
C --- ELECTRIC FIELD INTERACTIONS WITHIN FINITE ARRAYS AND THE DESIGN   FIN      9
C --- OF WITHDRAWAL WEIGHTED SAW FILTERS AT FUNDEMENTAL AND HIGHER      FIN     10
C --- HARMONICS.                                                        FIN     11
C --- BY LAKER, COHEN, & SLOBODNIK,JR.  1976 ULTRASONICS SYMPOSIUM      FIN     12
C --- PROCEEDINGS, IEEE CAT. NO. 76  CH1120-5SU, PP.317-321.            FIN     13
C                                                                       FIN     14
C --- A TRAPEZOIDAL RULE IS USED WITH AN FFT ROUTINE TO EFFECT THE      FIN     15
C --- DOUBLE INTEGRATIONS.                                              FIN     16
C                                                                       FIN     17
C                                                                       FIN     18
C                                                                       FIN     19
C --- USER SUPPLIES (IN NAMELIST)                                       FIN     20
C     1.  ETA = ELECTRODE WIDTH/CENTER-TO-CENTER SEPARATION             FIN     21
C     2.  NP = NO. OF POINTS IN THE QUADRATURE ( = 128, USUALLY)        FIN     22
C     3.  NN = NO. OF TERMS IN THE CHARGE DENSITY EXPANSION ( = 6)      FIN     23
C     4.  ITOTAL= MAX. NO. OF ELECTRODES IN ARRAY                       FIN     24
C                                                                       FIN     25
C --- ARRAY DIMENSIONS                                                  FIN     26
C     Y(NP)                                                             FIN     27
C     YN(NN,NP)                                                         FIN     28
C     PSI(NN,NN,2*ITOTAL-1)                                             FIN     29
C                                                                       FIN     30
C --- OFF-DIAGONAL ELEMENTS ARE STORED IN PSI(X,X,X)                    FIN     30
C                                                                       FIN     31
C                                                                       FIN     32
C                                                                       FIN     33
      READ PARAMS                                                       MAIN     7
      PRINT PARAMS                                                      MAIN     8
      PI=3.14159265358979                                               MAIN     9
      JTOP=2*ITOTAL-1                                                   MAIN    10
      NP2=NP/2                                                          XX       3
      NP2P1=NP2+1                                                       XX       4
      NUP=NP2P1+1                                                       XX       5
      J=ITOTAL                                                          XX       6
      ALF=2.*PI/FLOAT(NP)                                               MAIN    11
      DTDPHI=ALF*ALF                                                    MAIN    12
      DO 10 KMJP7=1,JTOP                                                MAIN    13
      IF(KMJP7.EQ.J)GO TO 10                                            MAIN    14
      ETAKJ=2.*FLOAT(KMJP7-J)/ETA                                       MAIN    15
      DO 20 IO=1,NP                                                     MAIN    16
      PHIO=2.*PI*FLOAT(IO-1)/FLOAT(NP)                                  MAIN    17
      CALL YFFORM                                                       MAIN    18
      CALL FOUPT(Y,NP,1,1,-1,0)                                         MAIN    19
      DO 30 N=1,NN                                                      MAIN    20
   30 YN(N,IO)=REAL(Y(N))                                               MAIN    21
   20 CONTINUE                                                          MAIN    22
      DO 40 N=1,NN                                                      MAIN    23
      CALL XCFORM(N)                                                    MAIN    24
      CALL FOUPT(Y,NP,1,1,-1,0)                                         MAIN    25
      DO 50 I=1,NN                                                      MAIN    26
```

Program SINGLE computes the acoustic response for one or more electrod in a single electrode array. The user must supply the following:

A. in NAMELIST/PARAMS/
   1. NN (Same as in GAL3SE documentation)
   2. NOMEGA = no. of acoustic response ordinates
      = 480 for 0 f/f_0 12
   3. JTH (Same as in GAL3SE documentation)
   4. NCASE " " " " "
   5. ITOTAL " " " " "

B. in NAMELIST/VSEQ/
   1. NUM (Same as in GAL3SE documentation)
   2. JTH " " " " "
   3. POL " " " " "

ARRAY DIMENSIONS

AMPL(NOMEGA)
PHSE(NOMEGA)
SEQ(ITOTAL)
C(NN)
POL(ITOTAL)

BCD(8) = title or header card

Program DOUBLE is similar but the dimension of C must be at least NN*ITOTAL.

```
      PROGRAM SINGLE(INPUT,OUTPUT,TAPE4,TAPE8,TAPE9,PUNCH)
      COMMON AMPL(960),PHSE(960),NOMEGA,DOMEGA,SEQ(21),BCD(8),LS,LSHIFT
      COMMON NN2,PI,TWOPI,MARG,C(6)
      DIMENSION POL(21)
      DIMENSION TABLE(64,2)
      DIMENSION IBASE(7)   ,IPOL(7),KPOL(7)
      DATA IBASE/1,3,9,27,81,243,729/
      DATA HA/1HA/
      NAMELIST/PARAMS/NN,NOMEGA,JTH,NCASE,ITOTAL
      NAMELIST/VSEQ/NUM,JTH,POL
C
C
C
C
C
C
C
C   ***    THIS PROGRAM COMPUTES THE ACOUSTIC RESPONSE FUNCTION FOR
C   ***    ONE ELECTRODE AT A TIME IN SINGLE ELECTRODE ARRAY.
C   ***    THE RESPONSE COMPUTED BY THIS PROGRAM IS THE PRODUCT OF THE
C   ***    FOURIER TRANSORM OF THE NORMAL D-FIELD UNDER THE ELECTRODE
C   ***    IN QUESTION AND A FREQUENCY-DEPENDENT SKEWING TERM.
C   ***    THE NORMAL D-FIELD WAS OBTAINED USING THE STORED VALUES
C   ***    FROM TAPE4 OF THE SURFACE CHARGE DENSITY EXPANSION COEFFS.
C   ***    COMPUTED BY PROGRAM GAL3SE
C
C
C
   30 FORMAT(8A10)
  149 FORMAT(/45X,21(A1,2X)/)
  150 FORMAT(1H1,20X,*MAGN  OF FREQ  RESPONSE*/(1X,10E12.4))
  151 FORMAT(30X,5F13.10)
  160 FORMAT(//20X,*PHASE SPECTRUM*/(1X,10F10.6))
  198 FORMAT(//45X,*CONFIGURATION NO.  *,I3,10X,*ELECTRODE NO.  *,I2,2X,
     1    *INDEX=*,I2/)
  350 FORMAT(//45X,*EXPANSION COEFFICIENTS*/(1X,10F13.10))
  439 FORMAT(1H1,20X,*HARTMANN TABLE*///)
  440 FORMAT(10X,8F10.6/10X,8F10.6///)
  675 FORMAT(//45X,*CASE NO *,I2)
C
      READ PARAMS
      PRINT PARAMS
C
      DOMEGA=.019634954
      NOPRINT =1
      NOPLOT=1
      LS=40*JTH+1
      DENOM=24.*DOMEGA
      PI=3.14159265358979
      TWOPI=2.*PI
      ITM2=ITOTAL-1
C
C        FOR TESTS, SET
      JJ=11  $  KSTRIP=11
      ITM2=(ITOTAL+1)/2
C
      NN2=NN/2
C
      READ 30,BCD
```

```
C
      DO 200 LA=1,NCASE
      READ(4,25)KASE,JJ,POL,C
   25 FORMAT(10X,I3,10X,I2/3(5X,7F5.0/),6F10.6)
      NUM=KASE
      IF(JJ.NE.11)GO TO 200
      CALL AMPPHSE
C
      LS=40*JTH+1
   22 CONTINUE
C
      IF(NOPRINT.EQ.1)GO TO 657
      PRINT 52,(AMPL(KI),KI=1,NOMEGA)
   52 FORMAT(10X,*SPECTRAL RESPONSE AMPLITUDES*//((1X,10F11.4)))
      PRINT 160,(PHSE(I),I=1,NOMEGA)
  657 CONTINUE
      IF(NOPLOT.EQ.1)GO TO 445
      K2=KSTRIP
      ENCODE(22,455,BCD(5))KASE,K2
  455 FORMAT(4H NO.,I2,1X,10H ELECTRODE,I3)
C     CALL DZPLOT
  445 CONTINUE
C
C     PHASE-SLOPE USING 5-POINT FORMULA
C
C ----    THE DERIVATIVE IS WITH RESPECT TO NU, AND IS EXACTLTY
C ----    FORMULATED IN TERMS OF FINITE DIFFERENCES. ------------
C
      PS=2.*PHSE(LS-2)-16.*PHSE(LS-1)+16.*PHSE(LS+1)-2.*PHSE(LS+2)
      PS=-.125*PS/DENOM
C     ----      INTEGERIZING OF POLARITY SEQUENCE AND CREATION
C     ----      OF THE SHIFTED BASE 3 INDEX = LOC.
C
      DO 107 I=8,14
      II=I-7
  107 IPOL(II)=INT(POL(I))+1
      KPOL(1)= IPOL(4)
      KPOL(2)=IPOL(5)
      KPOL(3)=IPOL(6)
      KPOL(4)=IPOL(7)
      KPOL(5)=IPOL(1)
      KPOL(6)=IPOL(2)
      KPOL(7)=IPOL(3)
      LOC=1
      DO 110 I=1,6
      II=8-I
      LOC=LOC+KPOL(II)*IBASE(I)
  110 CONTINUE
      POL(11)=-1.
      WRITE(8,VSEQ)
      IIA=LOC  $  IIB=1  $  IIC=2
C     PUNCH 45,IIA,IIB,IIA,IIC,AMPL(LS),PS
   45 FORMAT(7X,*DATA TABLE(*,I3,*,*,I1,*), TABLE(*,
     $   I3,*,*,I1,*)/*,F9.6,*,*,F9.6,*/*)
      PRINT 95,KASE,(POL(K),K=8,14)
   95 FORMAT(5X,I4,5X,7F4.0)
   36 FORMAT(5X,*CASE NO.  =*,I4,5X,7F4.0,5X,2(1X,F9.6))
C     PRINT 35,KASE,(POL(K),K=8,14), (IPOL(K),K=1,7),(KPOL(K),K=1,7),
C    $        LOC,AMPL(LS),PS
   35 FORMAT(1X,I3,3X,7F4.0,3X,7I4,3X,7I4/10X,I5,1X,2(1X,F9.6)//)
C
  200 CONTINUE
C
```

```
      COMMON NN2,PI,TWOPI,MARG,C(6)
      DATA D/1.E-9/
C
C  ---  NOTE.  THE PHASE CALCULATED HERE IS THE NEGATIVE OF
C  ---         THE PHASE OF THE ACOUSTIC RESPONSE
C
      TWOPI=0.
      LA=LS-2
      LB=LS+2
      DO 10 LS= LA,LB
      OMEGA=.25*PI*FLOAT(LS-1)/40
      II=LS
      L=II
      SUMR=SUMI=0.
      DO 100 NT=1,NN2
      NM=NT-1
      NE=NM*2
      NO=2*NM+1
      CALL BESJ(OMEGA,NE,BJ,D,IER)
      IF((OMEGA.EQ.0.).AND.(NE.EQ.0))BJ=1.
      EYE=COS(PI*FLOAT(NM))
      NA=NO
      SUMR=SUMR+C(NA)*BJ*EYE
      CALL BESJ(OMEGA,NO,BJ,D,IER)
      IF((OMEGA.EQ.0.).AND.(NO.EQ.0))BJ=1.
      EYE1=-EYE
      SUMI=SUMI+C(NA+ 1)*BJ*EYE1
  100 CONTINUE
      SQ=SUMR*SUMR+SUMI*SUMI
      SQ=SQ*4.*DOMEGA*FLOAT(II-1)/PI
      IF(SQ.LE.0.)PRINT 25,SQ,SUMR,SUMI,DOMEGA,II
   25 FORMAT(10X,*SQ=*,E11.4,1X,*SUMR=*,E11.4,1X,*SUMI=*,E11.4,1X,
     $  *DOMEGA=*,E11.4,1X,*II=*,I4)
      T=SQRT(SQ)
      AMPL(L)=T/.1905824111
      PHSE(L)=0.
      IF(SUMR.NE.(0.).AND.SUMI.NE.(0.))PHSE(L)=ATAN2(SUMI,SUMR)
   50 CONTINUE
   10 CONTINUE
      RETURN
      END
      PROGRAM DOUBLE(INPUT,OUTPUT,TAPE6,TAPE9)
      COMMON AMPL(960),PHSE(960),NOMEGA,DOMEGA,SEQ(22),BCD(8),LS,LSHIFT
      COMMON NN2,PI,TWOPI,MARG,C(190)
      DIMENSION IONE(8),ITWO(8),PABLE(64,2),IDD(8)
      DIMENSION TABLE(64,2)
      DATA IONE/1,9,16,22,27,31,34,36/
      DATA ITWO/8,15,21,26,30,33,35,36/
      DATA IDD/0,1,3,6,10,15,21,28/
      DATA HA/1HA/
      NAMELIST/PARAMS/NN,NOMEGA,DOMEGA,NORMOP,NOPLOT,NOPRINT
     1    ,JTH,NCASE,ITOTAL
C
C
C
C
C  THIS PROGRAM COMPUTES AND PLOTS THE ACOUSTIC RESPONSE FUNCTION FOR
C  AN ELECTRODE PAIR IN AN ARRAY OF ELECTRODES.
C  THE RESPONSE IS THE PRODUCT OF  THE FOURIER TRANSFORM OF THE
C  NORMAL D-FIELD UNDER THE PAIR IN QUESTION AND A FREQUENCY
C  DEPENDENT SKEWING TERM.
C  THE NORMAL D-FIELD WAS OBTAINED USING THE STORED VALUES OF THE
C  APPROPRIATE SURFACE CHARGE DENSITY COEFFICIENTS WHICH WERE COMPUTED
C  BY ANOTHER PROGRAM (GALERKN-TYPE)
C
C
C
C
   30 FORMAT(8A10)
  149 FORMAT(/45X,22(A1,2X)/)
  150 FORMAT(1H1,20X,*MAGN.  OF FREQ.  RESPONSE*/(1X,10E12.4))
  151 FORMAT(30X,5F13.10)
  160 FORMAT(//20X,*PHASE SPECTRUM*/(1X,10F10.6))
  198 FORMAT(//45X,*CONFIGURATION NO.  *,I3,10X,*ELECTRODE NO.  *,I2,2X,
     1    *INDEX=*,I2/)
  350 FORMAT(//45X,*EXPANSION COEFFICIENTS*/(1X,10F13.10))
  439 FORMAT(1H1,20X,*HARTMANN TABLE   -   FOR F/FO=*,I2///)
  440 FORMAT(10X,8F10.6/10X,8F10.6///)
  675 FORMAT(//45X,*CASE NO.*,I2)
C
      MM=0
      READ PARAMS
      PRINT PARAMS
C
      LS=40*JTH+1
      DENOM=24.*DOMEGA
      PI=3.14159265358979
      TWOPI=2.*PI
      ITM2=ITOTAL-1
C
C
      K2=6
      NN2=NN/2
C
      READ 30,BCD
      PRINT 30,BCD
C
      DO 200 LA=1,NCASE
      READ(6,198)KASE,JJ,KM
      READ(6,149)(SEQ(I),I=1,ITOTAL)
      READ(6,151)(C(N+MARG),N=1,NN)
      PRINT 198,KASE,JJ,KM
      PRINT 149,SEQ
      PRINT 151,(C(N+MARG),N=1,NN)
      LSHIFT=0
      CALL AMPPHSE
C
   22 CONTINUE
      READ(6,198)KASE,JJ,KM
      READ(6,149)(SEQ(I),I=1,ITOTAL)
      MARG=(KM-1)*NN
      READ(6,151)(C(N+MARG),N=1,NN)
      CALL ERRSET(MM,20)
      PRINT 198,KASE,JJ,KM
      PRINT 149,SEQ
      PRINT 151,(C(N+MARG),N=1,NN)
      LSHIFT=NOMEGA
C
      CALL AMPPHSE
      CALL PAIR
C
      IF(NOPRINT.EQ.1)GO TO 657
      PRINT 52,(AMPL(KI),KI=1,NOMEGA)
   52 FORMAT(10X,*SPECTRAL RESPONSE AMPLITUDES*//((1X,10F11.4)))
      PRINT 160,(PHSE(I),I=1,NOMEGA)
  657 CONTINUE
      IF(NOPLOT.EQ.1)GO TO 445
      ENCODE(20,455,BCD(5))KASE,K2
  455 FORMAT(4H NO.,I3,10H,    PAIR #,I3)
      CALL DZPLOT
  445 CONTINUE
C
C     PHASE-SLOPE USING 5-POINT FORMULA
      PS=2.*PHSE(LS-2)-16.*PHSE(LS-1)+16.*PHSE(LS+1)-2.*PHSE(LS+2)
      PS=-.125*PS/DENOM
      TABLE(LA,1)=AMPL(LS)
      TABLE(LA,2)=PS
C
  200 CONTINUE
C
      A1=TABLE(1,1)
      DO 250 LA=1,64
  250 TABLE(LA,1)=TABLE(LA,1)/A1
      DO 329 II=1,8
      ID=IDD(II)
      I1=IONE(II)
      I2=ITWO(II)
      DO 330 I=I1,I2
      FABLE(I+ID,1)=TABLE(I,1)
  330 FABLE(I+ID,2)=TABLE(I,2)
  329 CONTINUE
      DO 285 II=1,8
      III=8*(II-1)+1
      IV=III+7
  285 CONTINUE
      PRINT 439,JTH
      DO 328 IROW=1,8
      DO 327 ICOL=1,8
      IT=8*(IROW-1)+ICOL
      JT=8*(ICOL-1)+IROW
      IF(ICOL.LT.IROW)GO TO 327
      FABLE(JT,1)=FABLE(IT,1)
  327 FABLE(JT,2)=FABLE(IT,2)
  328 CONTINUE
```

```
      DO 295 II=1,8
      III=8*(II-1)+1
      IV=III+7
      PRINT 440, (FABLE(J, 1), J=III, IV), (FABLE(J, 2), J=III, IV)
      WRITE(9, 440) (FABLE(J, 1), J=III, IV), (FABLE(J, 2), J=III, IV)
  295 CONTINUE
      CALL ENDPLT
      END
      SUBROUTINE AMPPHSE
      COMMON AMPL(960), PHSE(960), NOMEGA, DOMEGA, SEQ(22), BCD(8), LS, LSHIFT
      COMMON NN2, PI, TWOPI, MARG, C(190)
      DATA D/1. E-9/
C
      DO 50 II=1, NOMEGA
      SUMR=0.   $   SUMI=0.
      OMEGA=. 5*DOMEGA*FLOAT(II-1)
      DO 100 NT=1, NN2
      NM=NT-1
      NE=NM*2
      NO=2*NM+1
      CALL BESJ(OMEGA, NE, BJ, D, IER)
      IF((OMEGA. EQ. 0. ). AND. (NE. EQ. 0))BJ=1.
      EYE=COS(PI*FLOAT(NM))
      NA=NO+MARG
      SUMR=SUMR+C(NA)*BJ*EYE
      CALL BESJ(OMEGA, NO, BJ, D, IER)
      IF((OMEGA. EQ. 0. ). AND. (NO. EQ. 0))BJ=1.
      EYE1=-EYE
      SUMI=SUMI+C(NA+ 1)*BJ*EYE1
  100 CONTINUE
      SUMR=-SUMR
      SUMI=-SUMI
      SQ=SUMR*SUMR+SUMI*SUMI
      L=II+LSHIFT
      T=SQRT(SQ)
      AMPL(L)=T
      PHSE(L)=0.
      IF(SUMR. NE. (0. ).  AND. SUMI. NE. (0. ))PHSE(L)=ATAN2(SUMI, SUMR)
      IF(PHSE(L). LT. (0. ))PHSE(L)=PHSE(L)+TWOPI
   50 CONTINUE
      RETURN
      END
      SUBROUTINE PAIR
      COMMON AMPL(960), PHSE(960), NOMEGA, DOMEGA, SEQ(22), BCD(8), LS, LSHIFT
      COMMON NN2, PI, TWOPI, MARG, C(190)
C
      DO 10 I=1, NOMEGA
      II=I+NOMEGA
      FI=FLOAT(I-1)/40.
      ARG=PHSE(I)-PHSE(II)+. 5*PI*FI
      PHIM=PHSE(II)-FI
      PHIP=PHSE(I)+FI
      SUMR=AMPL(I)*COS(PHIP)+AMPL(II)*COS(PHIM)
      SUMI=AMPL(I)*SIN(PHIP)+AMPL(II)*SIN(PHIM)
              T=AMPL(I)**2. +AMPL(II)**2. +2. *AMPL(I)*AMPL(II)*COS(ARG)
      BB=2. *DOMEGA*FLOAT(I-1)/PI
      TT=SQRT(T*BB)
      AMPL(I)=TT/. 1287985
      IF(AMPL(I). GE. (2. 13))AMPL(I)=2. 13
      PHSE(I)=0.
      IF(SUMR. NE. (0. ). AND. SUMI. NE. (0. ))PHSE(I)=ATAN2(SUMI, SUMR)
   10 CONTINUE
C
      AMPL(1)=0.
      END
      SUBROUTINE DZPLOT
      COMMON AMPL(960),PHSE(960),NOMEGA,DOMEGA,SEQ(22),BCD(8),LS,LSHIFT
      COMMON NN2,PI,TWOPI,MARG,C(190)
      DIMENSION XA(480)
      DIMENSION POL(19)
      XSIZE=12  $  YSIZE=8.
C
C
C     NORMALIZATION OF SPECTRUM
C
      HITE=.1
      PRINT 50,(AMPL(L),L=1,NOMEGA)
   50 FORMAT(1H1,20X,*SPECTRAL RESPONSE FUNCTION*//((1X,10F11.4)))
C     AMPLITUDE SPECTRUM PLOT
      CALL PLTID3(26HE. COHEN  **DOUBLE** X2402,30.,11.,1.0)
      DO 40 L=1,NOMEGA
   40 XA(L)=4.*DOMEGA*FLOAT(L-1)/PI
C     X-AXIS
      DVX=10.
      CALL SCALE(XA,12.,NOMEGA,1,DVX,XMIN,DX)
      CALL AXIS(0.,0.,5H F/FO,-5,XSIZE,0.,XMIN,DX,DVX)
      XC=3.0
      YC=8.7
      CALL SYMBOL(XC,YC,HITE,BCD,0.,80)
      ENCODE(54,55,POL(1))(SEQ(I),I=1,22)
   55 FORMAT(10X,22(A1,1X))
      YC=8.5
      CALL SYMBOL(XC,YC,HITE,POL,0.,80)
C     Y-AXIS
      DVY=10.
C----- YMAX = MAX. ORDINATE VALUE
C-----YMIN = MIN. ORDINATE VALUE
C-----YNUM = NUMBER OF INTERVALS COMPOSING (YMIN,YMAX)
C-----YLNGTH = LENGTH OF Y-AXIS (INCHES)
      YMAX=2.  $  YMIN=0.  $  YNUM=20.  $  YLNGTH=8.
      DVY=10.*YNUM/YLNGTH
      DY=(YMAX-YMIN)/YLNGTH
      CALL AXIS(0.,0.,25HNORMALIZED AMP. RESPONSE,25,8.4,90.,YMIN,DY,
     1   DVY)
      CALL LINE(XA,AMPL,NOMEGA,1,0,11,XMIN,DX,YMIN,DY,.08)
C     PHASE SPECTRUM PLOT
C     X-AXIS
      ORIGIN=XSIZE+2.
      CALL PLOT(ORIGIN,0.,-3)
      CALL AXIS(0.,0.,5H F/FO,-5,XSIZE,0.,XMIN,DX,DVX)
      YC=8.7
      CALL SYMBOL(XC,YC,HITE,BCD,0.,80)
      YC=8.5
      CALL SYMBOL(XC,YC,HITE,POL,0.,80)
C     Y-AXIS
      DVY=10.
      CALL SCALE(PHSE,8.,NOMEGA,1,DVY,YMIN,DY)
      CALL AXIS(0.,0.,14HPHASE SPECTRUM,14,YSIZE,90.,YMIN,DY,DVY)
      CALL LINE(XA,PHSE,NOMEGA,1,0,11,XMIN,DX,YMIN,DY,.08)
      CALL PLOT(20.,0.,-3)
      RETURN
      END
```

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

Printed by
United States Air Force
Hanscom AFB, Mass. 01731